

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Location-Based Digital Games Platform for Touristic Activities

Rodolfo Paulo Teixeira Tavares da Silva



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: António Fernando Vasconcelos Cunha Castro Coelho (Doutor)

February 2013

Location-Based Digital Games Platform for Touristic Activities

Rodolfo Paulo Teixeira Tavares da Silva

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Eugénio Oliveira (Doctor)

External Examiner: Leonel Morgado (Doctor)

Supervisor: António Coelho (Doctor)

February 13, 2013

Abstract

When visiting a new area it's interesting to investigate and acquire some knowledge about the place. Yet, it would be more appealing for the tourists to have a previous customized and interactive experience to explore that place.

This dissertation was made in partnership with 3Decide, which operates in the area of procedural modeling and augmented reality. The company is developing a new framework, named aLivePanoramics, which integrates 360° images and augmented reality.

The goal of the dissertation was to create a platform and find a game design that could run under the aLivePanoramics architecture. The framework allows the creation of tours which are composed by widgets, meaning that the game mechanics should had this in consideration. The solution should be as general as possible so that the platform is able to create multiple games distinct from each other. These created games should give the tourists the desire to go to the promoted places.

The produced platform allows the creation of adventure games where it is required to solve enigmas in order to evolve in the game. There are two types of user: the client that uses the platform to create games; and the final user that is who plays the games. The platform allows creating games where the client can add scenarios and objects. These objects can be used in the scenario or combined with other objects to unlock passages to get to the final goal defined by the client as well.

Overall the goals were achieved successfully since it is possible to create distinct games with different stories. In the end a small prototype was made as a proof of concept. This prototype was presented to some people and allowed doing some examination and improvement to the user interface. With the removal of some buttons and windows, the interface became more accessible and less interaction is needed by the user to get feedback. This enables the user to focus on the story instead of struggling with the user interface to understand the game. The prototype is a small story with four scenarios and the goal being the arrival on the fourth scenario. The passages from scenario to scenario are locked and show some dialogs, in order to give clues to the user. The user has to find some objects, such as a key and a shovel to unlock the passages and progress in the game. When the user arrives at the end is presented a final text congratulating the achievement of the goal.

Resumo

Quando se visita uma nova área é interessante investigar e adquirir algum conhecimento sobre o local. No entanto, seria mais atrativo para os turistas terem uma experiência prévia customizada e interativa para explorar esse local.

Esta dissertação foi feita em parceria com a 3Decide, que opera na área da modelação procedimental e realidade aumentada. A empresa está a desenvolver uma nova Framework, intitulada aLivePanoramics, que integra imagens de 360° e realidade aumentada.

O objetivo da dissertação era criar uma plataforma e encontrar um *design* de jogo que explorasse a arquitetura do aLivePanoramics. A Framework permite a criação de tours que são compostos por *widgets*, o que significa que a mecânica de jogo deve ter estes aspetos em consideração. Adicionalmente, a solução deveria ser o mais geral possível de modo a que a plataforma permita a criação de vários jogos distintos. Estes jogos devem ser aliciantes o suficiente para causar nos turistas o desejo de ir visitar os locais a ser promovidos.

A plataforma desenvolvida permite a criação de jogos de aventura onde é necessário resolver enigmas de modo a se poder evoluir no jogo. Existem dois tipos de utilizador: o cliente que usa a plataforma para criar jogos; e o utilizador final que é quem joga. A plataforma possibilita criar jogos onde o cliente pode adicionar cenários e objetos. Estes objetos podem ser usados no cenário ou combinados com outros objetos, para desbloquear passagens e chegar ao objetivo final, que também é definido pelo cliente.

No geral, os objetivos foram atingidos com sucesso, uma vez que, é possível criar jogos diferentes com histórias distintas. No final foi feito um pequeno protótipo como prova de conceito. O protótipo foi apresentado a algumas pessoas, o que permitiu fazer alguns testes e melhorar a interface gráfica. Com a remoção de alguns botões e janelas, a interface ficou mais acessível, causando com que o utilizador tenha de fazer menos interações de modo a obter *feedback*, possibilitando assim, com que este não perca o rumo à história devido à experiência de jogo ser demasiado difícil de entender. O protótipo consiste numa pequena história com quatro cenários em que o objetivo é chegar ao quarto cenário. As passagens de cenário para cenário estão bloqueadas e mostram diálogos, de modo a dar pistas para o utilizador saber a razão desse bloqueamento. O utilizador tem de encontrar objetos como uma chave e uma pá para desbloqueá-las e avançar no jogo. Quando o utilizador chega ao fim, é apresentado um texto final a felicitar a chegada ao fim.

Acknowledgements

The development of this dissertation wouldn't be possible without the help of several people.

I would like to give my appreciation to my supervisor António Coelho, for his guidance, support and ideas throughout the project.

I would like to thanks the support from the workers of 3Decide, in particular to Carlos Rebelo and André Cardoso for their help and experience during the development of the project.

I want to give acknowledgement to the designer Gonalo Machado, for the interface mockups and excellent work.

Without my family efforts it wouldn't be possible to be here today. I want to give a special thanks to my mother and my father for their constant support and motivation.

I would also like to thank my friends for their support.

Rodolfo Silva

Contents

Introduction	1
1.1 Motivation and Context.....	1
1.2 Problem Description.....	2
1.3 Goals	2
1.4 Methodology	3
1.5 Document Structure.....	3
Bibliographic Revision.....	5
2.1 Mobile Games	5
2.2 Location-Based Services.....	7
2.2.1 History of Location-Based Services.....	7
2.2.2 Locating Methods.....	7
2.3 Location-Based Games	9
2.3.1 Location-based Games Characteristics	9
2.3.2 Location-Based Games Issues.....	11
2.3.3 Examples of Location-Based Games	13
2.4 Electronic Games and Tourism	16
2.5 Existing Game Platforms	17
2.5.1 Adventure Game Studio	17
2.5.2 Visionaire Adventure Studio	19
2.6 Framework aLivePanoramics.....	20
2.6.1 Front-Office.....	21
2.6.2 Back-Office	25
2.6.3 Communication between front-office and back-office.....	26
2.7 Conclusions	26
Digital Games Platform for Touristic Activities.....	29
3.1 Methodology	29
3.2 Game Design Concept.....	30
3.2.1 Concepts.....	30
3.3 Story Mode.....	31

3.3.1	Assumptions and Dependencies.....	32
3.3.2	User Characteristics.....	32
3.3.3	System Architecture	33
3.3.4	Game Mechanics	33
3.4	Front-office	35
3.4.1	Functional requirements.....	35
3.4.2	Non-Functional Requirements	37
3.4.3	Use Cases	37
3.4.4	User Interfaces.....	41
3.5	Back-office.....	45
3.5.1	Functional Requirements.....	45
3.5.2	Non-functional Requirements	47
3.5.3	Use Cases	48
3.5.4	User Interface.....	54
3.6	Summary and Conclusions.....	57
Implementation		59
4.1	Process.....	59
4.1.1	Database	59
4.1.2	Communication between front-office and back-office.....	62
4.2	Results	62
4.3	Summary	66
Conclusions and Future Work.....		67
5.1	Goal Satisfaction	67
5.2	Future Work	68
References		69

List of Figures

Figure 2.2.1 - Assisted GPS	8
Figure 2.3.1 - Botfighters gameplay divided between text messages and a website	14
Figure 2.3.2 - Radar showing nearby items	15
Figure 2.3.3 - Picture of the game The Journey II	15
Figure 2.3.4 - Transition to AR-view	16
Figure 2.5.1 – 7 days skeptic, a game created with AGS	19
Figure 2.5.2 – Deponia, a game created with VAS	20
Figure 2.6.1 - Generic aLivePanoramis architecture	21
Figure 3.4.1- Front-office use case model	38
Figure 3.4.2 - Inventory use case diagram	38
Figure 3.4.3 - Notebook use case diagram	39
Figure 3.4.4 - Widgets use case diagram	39
Figure 3.4.5 – Dialogs use case diagram	40
Figure 3.4.6 – Options use case diagram	40
Figure 3.4.7 - Bottom bar story button and stories available menu	41
Figure 3.4.8 – Initial panel	42
Figure 3.4.9 – Inactive notes and inventory buttons	42
Figure 3.4.10 – Inventory panel and details	43
Figure 3.4.11 – Notebook panel and details	43
Figure 3.4.12 – Use and combination interface	44
Figure 3.4.13 – Scenario widget panel	44
Figure 3.4.14 – Character dialog panel	44
Figure 3.4.15 – Ending panel	45
Figure 3.5.1 –Story mode back-office use case model	48
Figure 3.5.2 – Characters use case diagram	49
Figure 3.5.3 – Stories use case diagram	49
Figure 3.5.4 –Introduction use case diagram	50
Figure 3.5.5 – Ending use case diagram	50
Figure 3.5.6 – Scenarios use case diagram	51
Figure 3.5.7 – Inventory items use case diagram	51

Figure 3.5.8 – Game elements use case module	52
Figure 3.5.9 – Events use case diagram	53
Figure 3.5.10 – Initial page of the story mode	54
Figure 3.5.11 – Mockup of the story page	55
Figure 3.5.12 – Add scenarios page mockup	55
Figure 3.5.13 – Scenario panel mockup	56
Figure 3.5.14 – Add navigation spot mockup	56
Figure 3.5.15 – Objects page mockup	57
Figure 4.1.1 – Previous virtual tour table	59
Figure 4.1.2 – Components of the Story mode database	60
Figure 4.2.1 – Beginning panel	63
Figure 4.2.2 – “Work Place” scenario and some interface	64
Figure 4.2.3 – Character panel that appears after trying to use the “Go to the pool” locked element	64
Figure 4.2.4 – The 3 elements in the inventory and a dialog after interacting with the “Wallet”	65
Figure 4.2.5 – Ending scenario	66

Abbreviations and Symbols

AGS	Adventure Game Studio
AR	Augmented Reality
DMO	Destination Management Organization
FEUP	Faculdade de Engenharia da Universidade do Porto
GPS	Global Positioning System
LBS	Location-Based Services
LBG	Location-Based Game
NPC	Non Player Character
VAS	Visionaire Adventure Studio
WAP	Wireless Application Protocol

Chapter 1

Introduction

Tourism is an important income at national level. It would be interesting to provide the tourists with custom experiences to promote touristic places. As such, those experiences should be interactive and in the same way attractive, so that the tourists feel the curiosity to visit those touristic places. Usually the aspiration of a tourist is either to relax, enjoy themselves, or to acquire knowledge about a location. The tourist population is a large one, and encompasses a lot of people with different tastes, cultures and experiences. It's very important to have these aspects in mind when building a system for touristic activities. If those aspects aren't taken in consideration, only a small portion of the touristic population will be satisfied with the outcome.

In the last few years mobile devices have highly increased their performance, communication ratios and display resolutions. With the massive spread of mobile devices with GPS, new opportunities arise to create innovative touristic services for the exploration of a territory.

This dissertation explores the potential of location-based digital games to promote the potential of certain touristic places. Location-based digital games, started appearing around the year of 2002, and take advantage of the users' physical location to generate or present content. As such, many of these games require the player to physically move to a certain location or place in order to play the game.

1.1 Motivation and Context

The idea emerged from a spin-off company of FEUP, named 3Decide that operates in the area of procedural modeling and augmented reality. The company is currently developing a framework entitled aLivePanoramics, which integrates Augmented Reality (AR) with 360 images.

Introduction

The increasing development of mobile devices with GPS and better resource capabilities, increases the possibility of creating new and interesting systems for touristic activities. Taking this into consideration, the main motivation for this project was to give the tourists the possibility to explore territories in an attractive way, before going to the place itself. This project aims to promote touristic places by giving the user (in this case the tourist) a location-based game, in which the user can interact and explore certain locations of a place.

The project must be flexible enough in a way that it can be extended to several places, for example, hotels, event houses, parks, landmarks, and other types of tourism.

1.2 Problem Description

The problem that emerged was the creation of a game design that allows understanding a touristic place and at the same time, integrate advertising services. The created games should motivate in users the desire to go to the promoted places.

One requirement of this project was the extension of the framework aLivePanoramics. Based on what the framework had to offer, the incorporation of additional functionalities were needed to explore the digital games for tourism. The framework works based on widgets which are one of the main elements that allow the creation of tours (circuits from scenario to scenario). The widgets are graphical interfaces in the tours, usually represented by icons, that when interacting with them, they execute a certain action with feedback. Because of that, it will be necessary a set of widgets that compose the mechanic of the game. The widgets should be customizable enough to make possible the creation of distinct games.

1.3 Goals

The main goal of this project is to develop a platform that allows the user to create games for tourism. With that in mind there are specific objectives to be accomplished:

- To create a concept and a game design that allows the user to create game tours
- To develop a client-server model, where the client is the front-office and the server is the back-office
- Each created game should have the possibility to contain multiple scenarios
- Each scenario can have a set of widgets. The widgets will be stored in the server and it is possible to add new widgets for a scenario

Introduction

- The framework must be extended to support the new widgets and other essential elements of the game
- For the management of the game preferences it is also needed to extend the existing back-office of the framework.
- Prototype a game for evaluation

1.4 Methodology

For the development of the game platform, a research was conducted, focused on location based digital games to learn the positive aspects and constraints. Later on, a study on the 3Decide framework aLivePanoramics was performed so that constraints and capabilities of the framework were understood and what influence can have for this project. As it was mentioned before, this framework integrates Augmented Reality and 360 images. Thus, each scenario is going to be composed by a 360 image, and when the user is in the corresponding place of that scenario the image is replaced by the actual view and augmented reality is going to be used instead. The games created are going to be based on a client-server model, where the client interacts with the server to get information. The information stored in the server are the tours, the widgets and some statistics.

The technologies used for the development of the platform are Unity3D with the package NGUI, used for the user interface, XML that have the information to create the tours and Yii Framework for the back-office. In the end and as a proof of concept a prototype game was implemented.

1.5 Document Structure

Besides the introduction this dissertation has 4 more chapters. In chapter 2 it's described the state of art, some important platforms and a description of the aLivePanoramics framework. Chapter 3 details the game concept and architecture of the Project. Chapter 4 documents the implementation and the obtained results. In chapter 5, the conclusions are presented, as well as some of the future work.

Chapter 2

Bibliographic Revision

This chapter shows the state of art and exposes some of the related works that have the more substantial matter for the development of this project. As so, it focuses on demonstrating the most significant aspects about mobile games and the essence behind Location-based services. It also has a great focus on location-based games, their characteristics and issues. Furthermore some examples of games are going to be displayed. The chapter also explains a little about the tourist population as well as some of the platforms that helped developing the platform. In the end is reserved a small part to talk about the aLivePanoramics framework and how it works.

2.1 Mobile Games

Mobile gaming goes back to the 70s Mattel's LED-based handhelds, which used LEDs to create graphics in a simple handheld device. In the meantime more recent devices were launched, such as Gameboy, which was quite successful, had distinct versions that stayed on the market for a fair time [1].

In 1999 the world would see subscription games, email based games, mobile games services and WAP games. In 2001 the first pay-per-locate GPS game arrived, and by 2003 the market had Head to Head multi-player games and mobile games being sold in cartridges, like in the mobile console / phone Nokia NGage. In 5 years the mobile gaming went from text messaging and pixel art to 3D and multi-player gaming [1].

In the last few years mobile devices have highly increase their performance, have faster communications and higher resolution displays and that had a great impact in our everyday life.

Mobile games have become more and more complex over the years and currently they resemble console and PC games of a few years back.

Bibliographic Revision

A good gaming experience requires a lot from the user interface. It should be convenient, reliable, and usable so that the player can concentrate on playing the game and enjoying it instead of struggling with the user interface. In addition, the game design itself has a huge impact on the gaming experience. If the rules or game world contains implausible features, the players can be easily offended or frustrated and quit playing the game [3].

Games differ from utility software in some key characteristics. In games, the purpose is to have fun and enjoy playing the game. Learning to play the game, solving problems, or discovering new things is part of that experience. Moreover, in a game, the players do not know in advance what to expect. Normally with utility software, users define their own goals and content that they will manage [3].

In the article “Playability Heuristics for Mobile Games” [3] are mentioned some heuristics with some detail for mobile games. Some of the initial heuristics that can be found in that article are:

- Don't waste the player's time
- Prepare for interruptions
- Take other persons into account
- Follow standard conventions
- Provide gameplay help
- Differentiation between device UI and the game UI should be evident
- Use terms that are familiar to the player
- Status of the characters and the game should be clearly visible
- The Player should have clear goals
- Support a wide range of players and playing styles
- Don't encourage repetitive and boring tasks

These heuristics are only in an initial phase, but seem quite important to have in mind when developing a mobile game. With a quick overview of these heuristics we can see that they take into account some mobile characteristics such as the fact that mobile phones are used for short periods for killing time, players rarely read manuals and carrying a paper manual around would be inconvenient.

2.2 Location-Based Services

Location-based services can be defined as services that use information on the geographical position of a mobile device to provide added value to the user. Usually the most common user activities are [4]:

- Orientation and localization
- Navigation
- Search
- Identification
- Event check

The two basic actions locating and navigating mainly rely on geospatial information. Searching, identifying and checking however, need a bigger variety of different information. Examples of additional information for these actions are traffic information, weather forecasts, or even personal information of the user [4].

Commercial location-based services that run on mobile devices, such as restaurant finders are becoming more established. At the same time, research into new location-based services continues to move forward, in areas such as way-finding, cultural tourism and games [2].

2.2.1 History of Location-Based Services

Since the 1970s, the U.S. Department of Defense has been operating the global positioning system (GPS), a satellite infrastructure serving the positioning of people and objects. It was originally conceived for military purposes, but in the 1980s the U.S. government decided to make the system data freely available for other industries worldwide. Since then, industries have taken up the opportunity to access the position data through GPS and use it to enhance their products and services [5].

KDDI launched the first mobile phones equipped with GPS in December 2001 which was the first global LBS. After that, the worldwide users started to replace the older phones with the new ones equipped with GPS allowing the LBS to expand [1].

2.2.2 Locating Methods

There are some methods that allow locating the position on a certain mobile device. The methods depend on the technology available on the country the device is at, the hardware of the device and the position of the device itself [1]. The location of the users' device must be estimated aligned with some known framework. This framework could be the locations of the

base stations of the device phone network or the satellites of the Global Positioning System (GPS). Each system will provide different levels of performance and capabilities.

Three locating methods were examined:

Using the mobile phone network

This method locates the Base Transceiver Station which the device is using, and through the location of the Station, it can locate the position of the mobile device. However the position of the device is inaccurate. The mobile device can be anywhere around the Station from 2 to 20 kilometers [1].

Satellites triangulation

The location of the device is established by comparing the times that the signals of different satellites take to reach the GPS receiver. To locate the receiver it needs at least 3 satellites. It's usually a good approach for locating devices since it has good accuracy. The range goes from 2 to 20 meters [1]. There are several factors that might introduce errors in GPS like clock inaccuracies, rounding errors, multipath and atmospheric effects, etc. Since the earth is also not a perfect sphere, GPS receivers generally look to four or more satellites to compute the precise location. To obtain an accurate position, assisted GPS systems can be used to overcome eventually limitations that occur by using the cellular connection to transmit remotely-collected satellite navigation data from the base station to the cellular device as it can be seen in figure 2.2.1 [6].

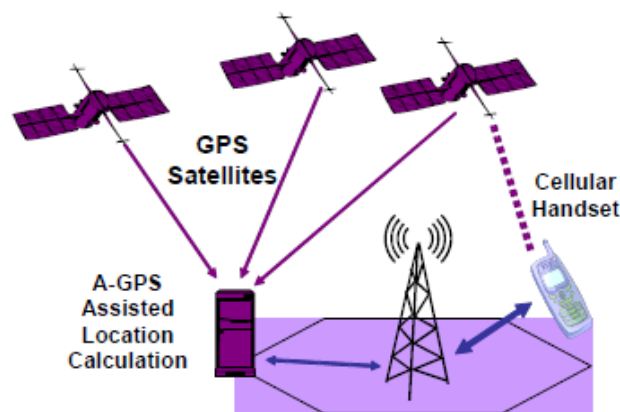


Figure 2.2.1 - Assisted GPS

Short range position beacons

This method uses the triangulation technique as well, but instead of using the satellites, it uses multiple Wi-Fi spots or Bluetooth. It's usually applied on small places such as malls. It is cheaper and more accessible [1].

2.3 Location-Based Games

A location-based game (LBG) is a game that uses the player physical location in order to play the game. The location of the player can be used to perform several events, such as the creation of levels or access location-specific information like maps, weather or even services. This type of game is mostly exclusively available for mobile platforms, due to the connectivity requirements that they require. Location-based games can also be called pervasive games, this is because it “implies the construction and enacting of augmented and/or embedded game worlds that reside on the threshold between tangible and immaterial space (...) in order to facilitate a “natural” environment for gameplay that ensures the explicitness of computational procedures in a post-screen setting” (Walther 2005). In other words, pervasive games get the player away from the PC and put them in the real world.

Location-based games can be simple, such as GPS drawing, where the player draws figures as he walks with the device, or more elaborated, like Geocaching, where the player has to solve puzzles and explore [1]. So, as we can see, LBGs can create different game experiences for the users.

2.3.1 Location-based Games Characteristics

LBGs present some challenges that other types of games don't, such as:

- Making people go outside
- Admitting that certain factors are beyond the games control
- The usage of the novelty of the location
- Keeping the games short and casual
- The existence of unexpected environmental factors
- Interacting with real world objects
- Negative perception by non-players
- Danger of breaking local rules

These challenges cause important aspects to have in mind in the development stage [6]:

- Movement is mostly designed as a necessity
- Games cannot simply be consumed. To entertain, they constantly need the players to act

Bibliographic Revision

- Games often require other users to function
- The ability of location-based games to handle player network latency in fast-moving games

Game developers often look at mobile devices as a constrained console platform and sometimes don't explore them to their fullest. This is unfortunate, as mobile devices can allow greater levels of interaction in both social and spatial dimensions, and so open up new possibilities for gaming. Location-based games are at an early stage in their evolution, and new cellular system capabilities and phone features are likely to significantly speed their future development. In LBGs is often to require the participation of other players in order to play the game. The reason many people play games or sports is that they wish to enjoy the experience with others, so this could be seen as a positive aspect [6].

There is often disagreement on what constitutes a story within a game. Some believe that a story should not merely provide some premise for action within a game but should evolve relative to a particular individual playing the game. Therefore many console and PC games can also be criticized for paying limited attention to the development of a good story. There have been some attempts to allow mobile players to engage in interactive stories, however, there is no evidence that mobile users want to play console-type games [6].

Latency can often be a hold back in Location-based games, especially when players are in close proximity and/or it's a fast moving game. Gameplay can be adjusted to take this in account, and the use of close range technologies such as Bluetooth could get around many of these problems.

Communication is also an important characteristic in LBGs. If the game needs the players to interact, since they aren't close enough to each other to communicate directly, the game must provide communication channels. Voice communication is a good feature, but if the bandwidth is low, text messages may be the only choice. A LBG can offer a range of targeting choices for messages. Apart from targeting specific players, a LBG can target players by location [7].

Security and rule violation become real issues when there is no direct eyesight between players. Contemporary online computer games come with a wide range of cheating counter-measures. Most violations can be blocked by the software before they happen. Location-based games don't have this privilege. The software that drives the game has no control over the physical world, so rule violations cannot be prevented. However, violations can be detected by the software and reported to the other players. This acts as both a deterrent and as an actual control instrument. The effect can be boosted with a tableau system that marks regular cheaters [7].

Massive multiplayer gaming comes quite naturally to location-based games. Since the game takes place in the real world, space is no constraint. There are no technical barriers, either. The mobile devices carried by the players constitute a distributed system. Thus there is a strong

incentive to use distributed systems algorithms to implement the game. One must pick an implementation that scales well to large numbers of players [7].

Further, as the majority of location-based games launched thus far have generated a considerable amount of public interest, they may well capture the interest of less traditional gamers [6].

LBGs can be divided in three categories with different degrees of dependence on location information [7]:

- **Mobile games:** In this kind of game, game events only occur when two players happen to meet. The game doesn't need to track the full position of the players. Proximity sensing and local communication are sufficient.
- **Location aware games:** In this kind of game, the geographical positions of the player matters. Game events can happen when a user visits a certain location.
- **Spatially aware games:** This kind of game integrates the real world surroundings of the players into the game. Buildings, roads and the landscape are available for use in the game. Game events occur when the player is in a certain spatial context.

2.3.2 Location-Based Games Issues

Usually software applications aren't perfect, and in the location-based games some issues arise as well. The most common problems that occur in LBGs are that they are unpredictable, sometimes information is unavailable or incomplete, and it's very hard to make an even gaming experience to all the players.

These issues can be categorized as [8]:

- Game-design issues
- Hardware limitations
- Location-related information availability
- Player's fitness
- Player's data protection

There are some design related issues that can emerge on this type of games. These issues mostly appear when the game mechanics are not correctly adapted to the location of the player. This causes a poor game experience even though the location of the player is being detected correctly.

Bibliographic Revision

Hardware limitations can be as well an important aspect in LBGs. LBGs depend on the GPS signal in order to know the player current position, that means that if this signal is unavailable, the game is unplayable. Other hardware limitations to have in mind are RAM, video memory or CPUs, because since LBGs usually are played in mobile devices which have a more limited capability.

Other issues that are present in LBGs are connected to location-related data information, such as maps, weather or other important hotspots on the area. If this information isn't available in the current location of the player the game becomes unplayable. Also, if the information is stored locally, it can eventually become outdated or limited to some locations, or if this information is accessed remotely via data connection means that the player has to pay for it.

The game should take in consideration the physical condition. Usually LBGs require the players to be active, which means that the more they play they will get more tired. If the game doesn't have that in account it will make the game harder to play.

To finish, there is the issue of privacy. Applications that allow the sharing of the user's private information are expected to handle that information with care and explicitly inform the user how is it being used and for what end. The game can as well access the user's location information that can be coordinates or the full address. This information is often cashed either for future use or statistical purposes. Nonetheless it must be handled with care.

These types of issues are common in location-based games and can keep away some potential players that don't feel comfortable in playing a game with some of these problems [8].

Solutions for Location-based Games Issues

Although some of these issues are regular, there are some solutions that can be performed in order to avoid them.

For the game-design issues it's important to know that the player is in a real world environment and when designing a game it's important to have that in mind and don't endanger the player in any possible way. If the gameplay is restricted we can make a game less unpredictable. The game-design is a vast area and to analyze it we would need to consider multiple case scenarios, but these issues can mostly be adjusted by a correct gameplay.

Hardware limitations are hard to overcome. In order to be playable, LBGs often require GPS or data connections and if the player is unable to get any of these services is incapable to play the game. However this can be prevailed if the game mechanics allow playing the game without using GPS, but still using some location as input (from logs or via direct input by the user). Of course these actions make the game experience poorer and although it would still be a LBG, it wouldn't be based on the real position of the player. There are some games that see the absence of a valid GPS signal as a hardware limitation and constrain the game to predefined locations to avoid negative experiences, however this limits the number of players that can play the game at all.

Bibliographic Revision

To solve information availability and suitability problems there are three possible ways: one is not use location related information at all, generating the needed information randomly, by using the player's GPS position as an input. This guarantees that the game can be played worldwide. The second alternative is storing all the data for the possible needed locations. Yet it's mostly likely that that information is going to get outdated especially if that information is temporary such as weather or traffic information. So this brings us to the third option that is the usage of remotely stored location-related content that is often updated via web services. However this implies the usage of data connections, limited location coverage, and possible remotely inaccessible servers.

As we know LBGs require the player to move in order to evolve in the game. This means that some games might be tiresome. So to solve this issue the game must take into account the player's fitness. Measuring the times that a player takes to go from one place to another is probably the best method to try to do an even game to all players so that the game is not extremely easy for some people and hard for others. Games should as well take into account that players might need to catch breath (when they stop the game can automatically stop as well). Basically to solve player's fitness issues when developing the game it should be considered that one person doesn't have infinite stamina.

For the privacy issues, the recommended approach is to store this private information somewhere locally or remotely. But since it's easy to lose a mobile phone or someone accesses it, the better option is to store this info remotely. If the game stores the information locally, for a better security this information should be protected by a username and password (for example), or the information showed should be minimized when the player is offline. Even so, the game should warn the player of the risks that are involved [8].

2.3.3 Examples of Location-Based Games

To have an idea what type of games there are in the market nowadays some games where investigated. There were other games that were seen, but I only placed here the ones that I thought were more significant. Nonetheless, in a general view, LBGs focus mainly in treasure hunting. Other common types in LBGs are competitive environments such as racing other players.

Botfighters

Botfighters was the first LGB to have a commercial use. The goal of the game was to destroy robots controlled by other players, being the player a robot as well [8]. When a target is destroyed the player earn some credits and advances on the high score list. As it can be seen in figure 2.3.1, there is a web interface where the player can spend the credits to improve the capabilities of his/her robot. The mobile phone is used for the battles on the streets [9].

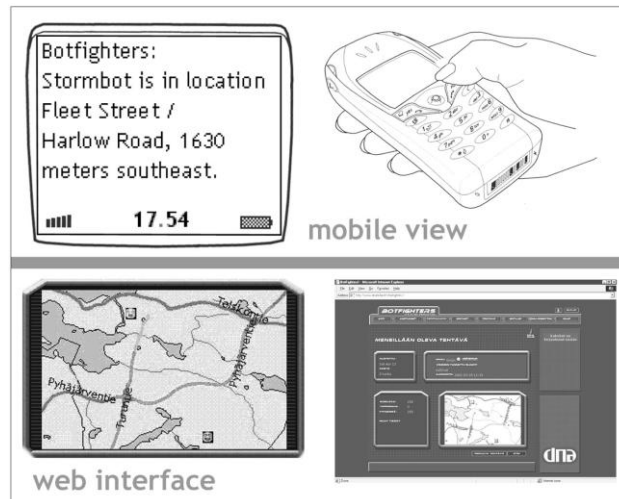


Figure 2.3.1 - Botfighters gameplay divided between text messages and a website

Mogi

Mogi is a treasure hunt game where the players use their cellular phones to collect objects. The aim of the game is to collect virtual treasure items that have been spread randomly over a virtual map of Tokyo, and the player needs to accumulate as many points as possible. Some items only appear at certain hours and places. In figure 2.3.2 we can see Mogis' radar that shows the nearby items in a certain area [6].



Figure 2.3.2 - Radar showing nearby items

The Journey II

This game attempts to introduce a more narrative story into an adventure game. The story indicates certain locations that the player has to go in order to solve mysteries. The game doesn't change the gameplay depending on the players current position, instead creates a virtual game environment as soon it realizes the player has moved. As we can see in the figure 2.3.3, a virtual bar is created where the player can interact with NPCs [6].



Figure 2.3.3 - Picture of the game The Journey II

Colonia Mysteria

Colonia Mysteria is a game where the story is around mythological Gods. Although some parts are fiction, the largest parts of its content are based on historical facts. After visiting the Roman-Germanic museum in Cologne, the players get game materials. The goal is to find treasures that will lead the players to Roman locations. After collecting 3 treasures, the players reach the target location where after solving some puzzles they get a key which allows seeing

the former ancient roman location overlapping the real world through AR. Figure 2.3.4 shows the transition from real world to AR-view [12].

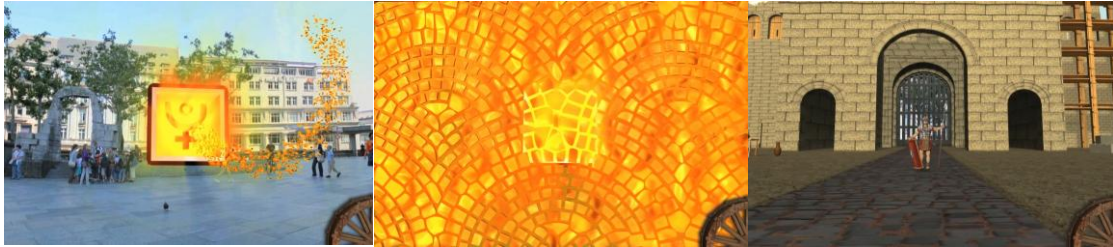


Figure 2.3.4 - Transition to AR-view

Geocaching

A famous LBG around the world is geocaching [13], which has millions of players worldwide. The goal is to locate hidden containers, called geocaches, using GPS-enabled devices. The geocaches are placed by players that give information regarding the location of the cache. Other players try to find that cache by exploring the area and solving riddles to determine the exact location [8].

2.4 Electronic Games and Tourism

This subdivision has the purpose to demonstrate some of the concerns that should be taken in consideration while developing games for a tourist population. The tourist population is a wide one, and when developing digital games for that population some particular aspects, such as attractiveness, accessibility and affordability are important.

When creating a game design for tourists it's important to have in mind that tourists usually are unfamiliar with the destinations they are visiting. This makes location-based services like navigation and orientation, more appealing. The location-based services are a great component that allows the tourists to understand the game physical boundaries, as well as helping to locate elements and landmarks relevant to the game in the real world. On the other hand, it is very important to select what events or encounters are a part of the game because it is significant that those contents add some excitement to the game experience so that the tourist doesn't get bored and leaves the game [15].

Since a tourist is usually in an unfamiliar environment, he may be willing to give up some private information to get access to better services. Of course that aspect depends on the temporary component of the visit. Nonetheless, the pursuing of privacy invasions should be taken to a minimum since it's a delicate subject [15].

Mobile devices play an important role on the tourist market and allow the usage of many touristic services. Mobile devices provide tourism companies, global distribution through

interactive user experiences long before customers arrive at the venue. Although mobile phones may be an important method for touristic services and games, shouldn't be the only focus. Many tourists may be reluctant or unable to use their own mobile phone because of roaming charges, internet issues or simply because they aren't comfortably enough to install a third party software in their personal phone since they don't know how, or are aware of potential malware threats [15].

According to the 2012 DMO Marketing Activities Study, destination management organizations (DMOs) are making significant investments in digital marketing, spending more than \$39 million on websites development over the last three years. This fact allows for the travel/tourism sales to occur online. Hotels have also invested in this area, meaning that a lot of them, support online sales for the customers. The tourism and travel companies in order to reach a global market are integrating social media applications to their marketing. This has many benefits over traditional online channels, such as cost-effectiveness, attraction for all groupings and demographics and allowing for the user to share experiences with friends. [16]

Other interesting elements are the digital brochures. If they present a professional appearance they can turn real-world platforms such as magazines or booklets into interactive and entertaining media. The virality of the content as well as the convenience of gathering all the travel data is very beneficial for the tourists [16].

As we can see the digital marketing is creating a great impact on the tourism and extending this subject matter into electronic games can only be beneficial, allowing the companies to advertise their service, but at the same time entertain the customers/tourists.

2.5 Existing Game Platforms

An important aspect of the project was to create a game platform for games. After deciding the concept (further analyses in the next chapter), two game platforms were studied to understand the concepts behind them.

2.5.1 Adventure Game Studio

Adventure Game Studio (AGS) is a game engine to create classic 2D point & click adventures. The whole development takes place in the AGS editor where the scene, walk-able & clickable areas as well as user interface and events are created. No programming is required to create simple adventures, however it is supported for advanced games [18].

To create a new game the user has to decide to give it a name that is displayed in the title bar when playing in the window mode. The main way to navigate around the editor is using the project tree in the top right corner, which allows accessing the various parts of the game. Before creating anything in the game the user can configure some settings of the game such as enabling

Bibliographic Revision

the debug mode, enabling anti-glide mode, if the click is pixel perfect, if the characters turn before walking, the color pallet and the resolution of the game.

In AGS the games are made up of several rooms, which the player moves between in order to play the game. Each room is one screen that has its own background, hotspots and objects. When creating a room the user must define some aspects, such as giving it a name, the background that is an image, the edges that define when the characters leaves the room, the walk-able areas and the walk-behind areas (when a room object is over the characters). The user can also define if the room is the starting point of the game.

There are three types of things the player can interact with and the user can add to the game: hotspots, objects and characters. A hotspot is an area of the background that the player can look at and interact with. An object is an item in the room that can move around the screen and be switched on and off. Objects must stay within the room that they are created in. A character is like an object except that it can move between rooms and appear all over the game. Since characters have talking interactions usually they represent NPCs.

Each character in the game can carry their own set of inventory items. Inventory items are totally separated from room objects and must be defined if the player can carry a certain item at a certain point in the game. The items have their own id and the user can define if the player starts with a certain item in the inventory.

Many of these components have events handlers where the user can personalize particular aspects of the components. For example, a hotspot has a “click” event, a “look at” event, a “mouse over” event, a “pick up” event, a “talk” event and others.

The user can import his own set of graphics if not satisfied with the default one and can do it by using the Sprite Manager. AGS uses sprites for all the game graphics, except room backgrounds. The Sprite Manager is the central place where the user can import the graphics. If the user wants to use images in the game (for mouse cursors, views, objects, etc), he can select the image to use from this manager.

The user can manage animations using views. A view is a set of one or more loops. A loop is a set of frames which, when played in sequence, give the illusion of movement. An animation can be configured in some aspects such as the delay of each frame, the sound of the animation and if it's flipped or not, meaning that the frame can be mirrored left-right or displayed normally.

The user can add cutscenes to the game. A cutscene is a sequence of actions that occur in the game that the player doesn't have control over. This is frequently used in introductions, as well as in other simpler in-game events.

AGS also supports the addition of conversations with characters. The user can create two types of dialogs: a simple one where the character has just a fixed line of dialogs, possible changing if talked to the same character again; or dialog trees trough which the player can choose the topics to talk about.

Bibliographic Revision

It is also possible to configure and import the cursors to obtain visual feedback of the possible actions available and the fonts in case the user isn't pleased with the default ones.

The figure 2.5.1 shows an interaction with a character from the game “7 days skeptic”, which was a game created with AGS.



Figure 2.5.1 – 7 days skeptic, a game created with AGS

2.5.2 Visionaire Adventure Studio

The Visionaire Adventure Studio (VAS) is similar to AGS since it is an engine that allows creating point & clicking adventures. Compared to AGS though, VAS is a bit cleaner and has more features like PNG support (AGS doesn't). It's similar to AGS as in the visual event system and defining walk-able and clickable areas [19].

Unlike AGS that require a bit of programming, VAS allows the users to create their own games without requiring extensive programming skills. Therefore, VAS provides a huge set of existing “Actions” that can be combined in many imaginary ways. This allows users to develop their own games without writing a line of code.

Just like in AGS the user can create characters, but it can be much more personalized like giving the character standing animations and giving the character different outfits. It is also possible to define up to 8 character angles. Mouse views and fonts customization are possible as well.

The user can configure what action the keyboard keys do, such as save, load or quit the game.

The scenes are the main component where the game takes place. The other components can be added to the scenes. It is also important to configure the way points in the scenes to know where the characters can move and the initial position and direction of the character in the scene. The user can also configure the character size in the various positions of the scenario (for example, to give a perspective illusion by making the character smaller at the end of the scene). It's possible to define scene objects and make interaction with them. The interaction can be personalized such as defining the position of the character when interacting, or if the character moves towards the object to interact with it. Besides that it is also possible to create teleporting

doors to go to another scene and to define the light map in the scenes. The light map makes the characters brighter or darker depending if he is on an area with light or shadow.

If the user wants he can add menus to the game. The menus work as a user interface that have buttons that execute a certain action (for example, save, load, exit and resume the game). The characters can as well have an inventory that can have different user interfaces for when open and closed. It is also possible to create items where can be placed in the scenes, inventory, combined with other items or even used in scene objects.

As in AGS it is also possible to create conversations with NPCs.



Figure 2.5.2 – Deponia, a game created with VAS

In the figure 2.5.2 it shows the game Deponia, which was created using the VAS engine, where the main character is having a conversation with an NPC.

2.6 Framework aLivePanoramics

As it is mentioned before, this project dealt with a framework called aLivePanoramics¹. This section has the purpose of explaining the thoughts behind this framework, the objectives of the same, as well as the multiple components involved around it.

The framework allows the creation, management and viewing of tours. These actions are handled by two different users, which are the clients and the final users. The clients are the ones that create and manage the tours while the final users can view the outcome of those tours. A tour is a trip with visits to a variety of places, which are composed by several elements that the final users can interact.

¹ The aLivePanoramics is still a Framework in development, which means some of the names of the elements presented in this document may be subject to changes

The framework can be divided in three major components, which are the graphical and view area (front-office) that is viewed by the final users, the back-office where the client can customize the preferences of the view, and the communication between them which is made through XML. The back-office communicates with the database where all the information is stored. In the figure 2.6.1 we can see how that architecture works.

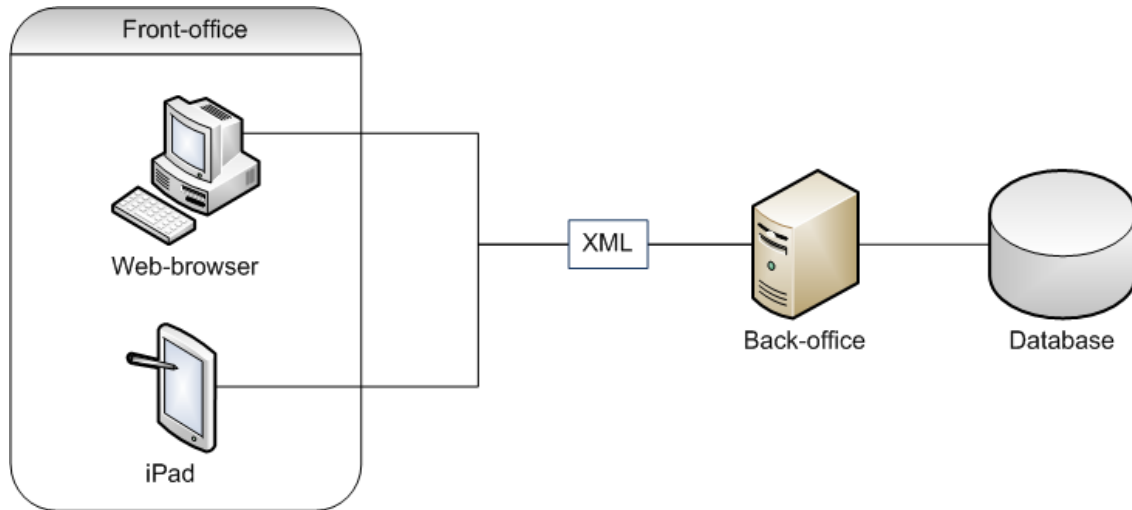


Figure 2.6.1 - Generic aLivePanoramis architecture

So, what are the thoughts and the purpose of this framework? Well, for starters one of the goals is to give the final users the possibility to explore a certain touristic location in a more interactive and interesting way. The final users can explore the location using Augmented Reality mode or using the Virtual Tour mode, which exploits pre-defined 360° images (panoramas). Both of these modes allow the final user to interact with the world/panoramas around him by clicking on widgets. On the other hand, the framework also provides the clients information about the final users, such as, what locations where the most liked in the social networks, which location is the most visited, and other statistics.

2.6.1 Front-Office

The front-office runs on an iPad or web browser and is the place where the final users can visualize the tours that were created by the clients. It receives a XML file from the back-office and interprets the data contained in it to build the corresponding tours. As it was mentioned before the application has two modes: the Virtual tours and the Augmented Reality. Although the purpose is for a mode to complement the other, the Virtual Tour mode and the Augmented Reality mode are distinct from each other. Both of these modes can be accessed from the main menu.

Virtual Tour:

The Virtual Tour mode is constituted by panoramas. These panoramas are 360° images that allow the user to view the touristic location as if he/she was there. This means that the user can see all the points of the panorama around him/her. To enter this mode the user has to choose one specific tour that appears in the list of the main menu.

A Virtual Tour is composed by several panoramas, each corresponding to a certain location in the world. Each panorama is then composed by widgets and can have an audio narration to explain a certain aspect of the place.

When the user enters the Virtual Tour mode, comes across with the first panorama of the tour, the widgets that belong to that panorama and a user interface. The interface provides the user a pleasant way of understanding the panorama and interact with it. The interface has many options that the user can interact.

In the Top Bar:

- **Back button:** allows the user to cancel the previous action
- **Home button:** allows to go to the main menu
- **Information button:** if there's some information about the panorama, this button is displayed and allows to see the that piece of information
- **Name of the panorama location:** Name of the current panorama
- **Search button:** allows to search for other panoramas in the tour
- **Clean screen button:** clears opened windows
- **Augmented reality button:** change to augmented reality mode

In the Bottom Bar:

- **Visualization buttons:**
 - **Auto button:** allows the application to control the visualization, and the camera moves around the panorama
 - **Gyro button:** when using the application in an iPad, when moving the iPad, the camera moves with it.
 - **Hand button:** allows the user to use the touch/mouse to move the camera
- **Audio buttons**

Bibliographic Revision

- **Play button:** plays the respective audio narration of the panorama, if there's one
- **Volume button:** allows to control the volume of the audio narration
- **Mute button:** no sound
- **Navigation buttons:** allows to navigate through the panoramas
- **Share button:** share the panorama on a social network
- **Captions button:** shows the captions for the audio narration

There are other interface elements on the screen:

- **Map button:** allows viewing the map of the tour with the several panoramas, and navigate to them.
- **Zoom buttons:** allows to zoom in or out the panorama view
- **Day/Night buttons:** switch from various time laps of the panorama
- **Radar:** allows to understand the distance of the other panoramas in the tour to the current panorama

Depending on the action of the widget, it may, or may not have a interface (for example, if a widget displays a video, the video is shown in a window, but if the widget is a navigation point, there isn't any interface, just a action that navigates to the corresponding panorama).

Augmented Reality:

In the Augmented Reality mode, the panoramas are replaced by the real world which is captured by the camera of the device. This mode only works on iPad and can be accessed in the main menu.

Keep in mind that this mode can be accessed when the user is in the location of a panorama and some of the widgets may vary.

As the name suggests, this mode adds to the reality an extra, in this case, the widgets. The allowed tour requires the user to move to a certain location in order to enjoy what it has to offer. The widgets have a defined location, and to interact with them, it's required that the device is in a certain range of that location. Since this mode is different from the Virtual Tour mode, the user interface is slightly distinct.

In the Top Bar:

- **Back button:** allows the user to cancel the previous action

Bibliographic Revision

- **Home button:** allows to go to the main menu
- **360° button:** change to Virtual Tour mode

Other interface elements on the screen:

- **Zoom buttons:** allows to zoom in or out the camera
- **View buttons:** allows to manage the distance of the viewing widgets
- **Map button:** allows to see the current position in the map

As we can see, the interaction in the Augmented Reality mode depends more in the present widgets of the current tour.

Widgets:

Both Virtual Tour and Augmented Reality modes can have widgets. And what is the potential of these widgets? What can they do? For now, as the framework is still in development, there aren't many widgets, but the goal is for the client to be allowed to do practically anything, such as, allowing the final user to share experiences on social networks, promote certain products or services, give information about a certain area, and other actions. Basically, the widgets are what make the product "alive" allowing the users to interact with the world and the panoramas.

The widgets are one of the most important elements in the application and can be divided in several packages:

- **Social widgets:** enable the interaction with social networks
- **Promotional widgets:** promote a certain service or product
- **Info widgets:** provide the information about something
- **Multimedia widgets:** these widgets can be any type of media, such as video, sound or text

There are other type of widgets that can only exist in one mode, such as the "navigation" widgets that allow to navigate from panorama to panorama in the Virtual Tour mode, and the "overwrite with an image" widget, that allows to see the current location in a older time by placing an image on top of the captured screen, in the Augmented reality mode. These are only a few examples, but there are many other.

2.6.2 Back-Office

To be able to interact and view the back-office the client needs to be authenticated first. The back-office is where the client can create and manage the Virtual Tour, and Augmented Reality modes. Besides that, the client can see the statistics and the XML files that are interpreted by the front-office. Each client can only see the corresponding XML files.

When the front-office runs in a web player, it is necessary to define the domain of the site in the back-office, in order to run properly.

Virtual Tour:

In the back-office the client can add new tours. Each tour is composed by a name and description. After the creation of a tour, there's the possibility to add new panoramas to it. A panorama has also got a name and description, can be added by uploading an image or by choosing an existing one from other tours. This means that a panorama can belong to several tours, and for each tour it can have a different name and description. The panoramas in one tour can be ordered according to the preferences of the client, which implies that the client doesn't have to worry about their order when adding new panoramas.

In the panoramas, besides the option to change the description, there's the possibility to configure the widgets, the audio narration that is played and the captions of that audio. There is also the possibility to configure the north position of the panorama.

When adding widgets, the client has to define a location in the panorama, and fill several fields depending on the widget (for example, if the widget is from navigation, it is necessary to define the target panorama to navigate).

Augmented Reality:

The Augmented Reality functionality in the back-office is where the client can add new widgets. These widgets have a real location coordinates that the client has to define. Like the widgets in the Virtual Tour mode, they also got other fields.

There isn't many detailed information about this functionality, since it's still in development.

Statistics:

Another functionality that is also in development is the statistics, but in this area of the back-office the client will be able to consult the information about the final users that used the application.

The information that the client will be able to consult, will be information such as:

- Likes of a certain tour or panorama
- The number of visits to a certain tour or panorama

- What number of users used a certain widget
- Other

With this information the client will be able to know more about their users and what they like, and can take action according to that.

2.6.3 Communication between front-office and back-office

The communication between the front-office and the back-office is made through a XML files. The XML files contain all the information about the virtual tours and augmented reality settings.

The virtual tour XML contains the tours of the user. Each tour is composed by the id, the title and the panoramas that belong to that tour. The panoramas then contain one id that represents them. Besides that, the info that the XML contains about the panoramas are: the name, the panorama preview and image sources, the audio sources, the captions, the coordinates of the panorama, the camera position, and the several widgets that compose the panorama. The info of the widgets varies accordingly to their action.

The creation of the XML files is made in the back-office. The back-office interprets the content of the tours and generates the XML files for each client. The client can see the content of these XML files trough his profile. In order to place the front-office in a web site, the client can get the deploy code in a javascript format, that can be placed in the corresponding site. For this to be able to work, the client has to define the site domain in the back-office.

2.7 Conclusions

When developing a game for mobile devices it must be taken in consideration many important aspects that aren't relevant for console or PC games. Some of those aspects are that the mobile games are usually played for short times and the user may want to save the progress. It is also important that the interface is appealing and easy to understand since the user doesn't have much time to study it.

LBGs can take advantage of many LBSs that the areas provide to increase the game experience. They usually require the users to go outside to play the game and interact with virtual or even real world objects. There are some issues that can emerge when developing LBGs but can also be surpassed if the right measures are performed. As we can see most of the games presented have a treasure hunting mechanics which is a very popular approach for location-based games. In the game Mogi it's interesting to see a mini map with the important items that are nearby, supporting the player with essential information so he/she can evolve in the game. The Journey II show us that it's not impossible to have a story in a location-based game. Colonia Mysteria shows that it's possible the integration of AR in location-based games

Bibliographic Revision

to enhance the game experience. With Geocaching we can see that many people don't inhibit themselves from moving from place to place, which is a good thing. Other interesting features that the games show is that people mostly like challenges and to overcome them in order that in the end there are some kind of reward, even if that reward is the top of a high score list.

The tourist population is a large one and since it covers a wide range of cultures it is important to be careful on the information displayed. The tourism companies are investing in sites and other applications which allow the exploration and growth of the area.

For point & click adventure game platforms it is important to make the scenes the main component where the other elements of the game are placed. It is also important to interact with objects of the scene and characters for the games to be interesting.

The aLivePanoramics framework is composed by three main components that are the front-office, the back-office and the communication between them. The front-office is the place where the final users can see the tours and interact with the widgets; the back-office is the place where the client can create and manage the tours and the files created by the back-office must be understood by the front-office. The framework is composed by panoramas (or locations) that have widgets that execute actions. The game mechanics must have in consideration these aspects since its going to work under this concept.

Chapter 3

Digital Games Platform for Touristic Activities

This chapter has the purpose of showing a detail presentation of the problem to be solved. The problem that was raised was to create a digital games platform, with the concept of the aLivePanoramics architecture. The platform should allow the creation of customizable games that allow getting some knowledge about the promoted touristic locations. Since the aLivePanoramics is a framework that already works with tourism, the target audience were the tourists. With these aspects in mind, there was the need to create a game design that is flexible, but attractive at the same time.

3.1 Methodology

The proposed goal for this project was to create a digital games platform for touristic activities. The solution should be as general as possible so that the platform is able to create multiple games distinct from each other. Both modes of the platform should be taken in consideration therefore the considered widgets should be general enough to work on both modes. Before the development of the platform itself, a study on the location-based games was performed as well as existing game platforms. These studies aimed to determine what issues these types of games have and what limitations had to be taken in consideration for the game design. Another important step was the study of the framework aLivePanoramics. Besides the study and observation of the framework, this step also includes the creation of an extension of the framework. This extension consisted on creating a new interactive widget called “Promo”. The widget had a percentage probability of appearing; it appeared in a random way (in a random panorama and in a random position of that panorama); when activated, the widget gave

a message and asked for the user to insert the mail; after inserting the mail, a final message appeared and the widget disappeared; the email should be reported to the server and stored in the data feed of the widget, so it is available in the back-office. The widget was successfully created and allowed to go through all the important aspects of the platform: the front-office, the back-office, the database, the XML, and the creation of a web-service. After the research was made, 3 game concepts were conceived. These concepts were analyzed and discussed with the team to perceive which one was the better option for the framework.

3.2 Game Design Concept

In the beginning of this project it was necessary to define a game design for the games that would be created by the platform. Since the framework aLivePanoramics used as main features the 360 images and widgets to control the environment, those were traits that were important to have in mind while thinking the game design. With those things in mind, three game design concepts arose. Those game concepts were treasure hunting, solving an enigma, and a questionnaire game. In this next section the main features and goals of those game designs are discussed.

3.2.1 Concepts

Treasure Hunting:

The player has to travel through several scenarios and find one or more objects on those scenarios in order to evolve in the game.

In the beginning the player is presented with a list of the objects that he has to find.

In order to create a bigger challenge to the player, when he finds an object he is presented with a mini puzzle that he has to solve if he wants unlock this object.

The player wins points according to the speed/number of moves which he makes to unlock the objects.

The obtained score can be submitted online to compete with other players.

Once an item is unlocked, besides the points obtained, it is also exposed to the player the information about the object or the scenario that the player is on.

Main widgets:

- **Objects:** Invisible (or not), that represent an object in the 360 image
- **Clues:** Contain information about the location of an object

Solve an Enigma:

The player has to travel through several scenarios to solve an enigma. To solve that enigma he has a number of objects that can combine in order to unlock other objects and evolve in the game.

The player can interact with objects in the scenarios (for example, a lever).

The player has an inventory that contains the objects collected which can be combined in order to create a new one.

The objects can be used in the scenarios in order to open other opportunities (for example, open a door to go to another scenario, or use a key in a drawer to find another object).

Main widgets:

- **Objects**
 - **Scenario Objects:** Can't be placed in the inventory
 - **Inventory Objects:** On click the object is removed from the scenario and it's placed in the players' inventory.
- **Clues:** Contain information about the next moves to be performed

Questionnaire Game:

In each scenario there are several widgets that test the players' knowledge about the touristic place.

In each scenario there are several widgets that correspond to a question.

The questions can be personalized.

There are widgets that exhibit images and the player has to answer something about that image.

There are widgets that replicate a sound and the player has to find out what instrument or object is playing that sound (for example).

The answers can be multiple choice or open response.

The player earns points according to the number of attempts he took in the questions.

The score can be submitted online.

3.3 Story Mode

After discussing the several concepts with the 3Decide team, a decision was made to go by the second approach, the "Solve an Enigma" concept, which enables the possibility to create stories. Allowing the clients to create their own stories about the touristic place appeared alluring, and a good way to attract people to play the game. After some thoughts about the game concept a decision was made to label this as "Story mode" in the framework.

As we have seen before, the Story mode is going to take the form of an adventure game where the user has to solve some enigmas that appear while he evolves through the game. The user can go to multiple scenarios where several objects are presented, and can combine and use those objects to solve the enigma.

The story mode is going to take advantage of the other modes of the framework aLivePanoramics and will be an extension to them. The story mode will be an extra, meaning that, if the client doesn't want a Story for his tours, it's not required to. A story can only have panoramas and locations that belong to the tour itself. As all the other modes in the framework, this mode has also the front-office and the back-office components.

It's expected that the usability of the Story mode is either unproblematic for both final users and clients. Since the final user population encompasses casual users, the interface should be easy enough to manage and give feedback without much effort. The back-office interaction should, as well, be simple enough that the client can focus on creating a story instead of trying to understand how to create one. Nonetheless, it should have enough content to create interesting stories.

3.3.1 Assumptions and Dependencies

The proposed solution works on a web-browser or an iPad and it requires Internet connection to be able to work. The information is stored in a database that is accessed by the front-office through web services that are provided by the back-office. If the user wants to take advantage of the AR mode it's necessary to turn on the GPS signal.

Needless to say, that the story mode must run under the aLivePanoramis framework to work properly.

3.3.2 User Characteristics

As in the aLivePanoramis framework, there will be two types of actors that interact with the Story mode: the final users and the clients. These actors have very distinct positions in the system and the actions they perform are in two different components.

Actors:

Final users: The final users are the ones that interact with the front-office and can be anyone that is looking to play a game related to a certain touristic location.

Clients: The clients are the ones that interact with the back-office. They can create new stories and manage them. To do this, first they need to be authenticated in the back-office.

3.3.3 System Architecture

The story mode has got the same architecture of the other modes. This means that it's composed by the following components:

- **Front-office:** the module where the final user can view and play the created stories
- **Back-office:** the module where the client can create new stories and manage them
- **XML communication:** the communication between the front-office and back-office is made through XML files
- **Database:** the back office communicates with the database

The architecture model can be seen in figure 2.6.1 in the bibliographic revision chapter.

3.3.4 Game Mechanics

This subdivision has the purpose to present in detail the game mechanics that the platform should provide. Since the game mechanics are the rules that produce the gameplay, most of the rules presented here are front-office oriented. Also keep in mind that this is a general solution to be able to create multiple game stories so, there aren't any specific details.

Visualization: Although it is possible to create many stories, when playing a game created by the platform there are some aspects that can't be configured. The visualization is one of those elements and it's in the first person, meaning that the user plays the role of the main character.

Inventory: The main character will have with him an inventory. In the beginning, the inventory starts empty and it's the place where he can visualize all the objects that are acquired during the story and can be used or combined.

Notebook: Besides the inventory, there is also a notebook that the main character owns and can visualize the obtained notes. The notebook also starts empty.

Beginning: When starting a new story the relevant information about that story is displayed to the user. The information that is shown is composed by the title of the current story, an introductory text and the goals of the user.

Scenarios: The scenarios will be the location where the stories take place and contain the game elements. Can be represented by panoramas or in case the user is using AR, the real world.

Game elements: The game elements are the widgets and are responsible for creating the enigma. The game elements can be: navigation spots, inventory objects, scenario objects, notes and characters. All these elements except the characters can be locked to prevent the user to go further and has to find a way to unlock them. All the elements can be visible or not.

Navigation spots: The navigation spots are what allow the user to go from scenario to scenario (they aren't available if the user is using AR). They have two different images for the locked state, this mean that if the navigation spot is locked, the image representing the closed navigation spot is shown, if not, the image representing it open is shown instead. This game element also has a name and a small description so that the user knows where it travels to.

Inventory objects: The inventory objects can be collected and placed in the inventory. When they are in the inventory they can be used in another game element to unlock it or can be combined with other inventory object to create a third one. The usage or combination will only work if in that story is specified to do so. The inventory objects can be used more than once if there is such need. They also have a name, a description and an image to represent them. If the inventory object is locked the image will appear in a grey scale to give visual feedback.

Scenario objects: A scenario object, unlike the inventory object, can't be collected and always stays in the scenario. When used, it unlocks another game element. These game elements have a name, a description and also an image to represent them. The image will be in a grey scale if the object is locked.

Notes: The notes will have the goal of giving the user some clues of what to do. They can also represent a book or other text information. They are represented by a sheet of paper when in the scenario and have a name, description and content when in the notebook. The sheet of paper will appear in a grey scale if the note is locked.

Characters: As we know the user takes the role of the main character, but there are other characters in the game the user can interact with. The main role of these characters is to give guidance to the user. They are represented by an image and a name.

Dialogs: There will be two types of dialogs, the ones that only give feedback and the interactive ones. The interactive dialogs are when the user can interact with another character by asking questions or simply talking (a quick example: the user selects "Hello, how are you?" from the possible dialogs options and the character replies "Hello, I'm fine thanks."). These questions are pre-defined by the client, as well as all the dialogs. The dialogs that only give

feedback appear on the screen and the user can't interact in any way with them. These dialogues can work as if it was the main character talking/thinking.

Triggers: A trigger occurs when an action is performed and has a consequence. Some examples of triggers that can happen are: after using a certain game element, another game element is revealed; when a game element is locked a dialog appears, giving the information on why the element is locked; when entering a new scenario a dialog is displayed.

Ending: When the game ends it is displayed a text congratulating the user. The ending conditions of a game can be when a user acquires a certain inventory item, or reaches a certain scenario. It is also possible to give the final user a reward, like a card. The final user can as well share the experience with his friends in a social network.

3.4 Front-office

The front-office is the module where the final users are able to play the games. In this subdivision of the document it will be detailed the requirements, the list of steps that define the interactions between the final user and the system, and the user interface mockups.

3.4.1 Functional requirements

There are many essential requirements for the front-office to work properly. Those requirements are listed below.

Access by the Virtual Tour mode: when the user is in the Virtual Tour mode he can access the Story mode if the current tour has related stories.

Save and load game: the user should be able to save the story and be able to proceed with it later on.

Scenarios: as it has been said before, the scenarios will be the panoramas or the real world.

Widgets: they represent the game elements, which contain a location in the scenario and, when the user interacts with them, a certain action is executed depending on the widget.

Navigation spot widgets: represent the navigation spots game elements. When the user interacts with them, a window appears showing the name and description, as well as a use button that allows traveling to the assigned scenario.

Inventory object widgets: represent the inventory objects while in the scenario and contain a certain inventory item. The widgets are represented by the corresponding item image. When the user interacts with them, the widgets disappear and the item is moved to the inventory.

Scenario object widgets: these widgets represent the scenario objects game elements. When the user interacts with them a window appears with the name and description of the object, as well as a button that allows using them.

Note widgets: represents the notes game elements. When the user interacts with them, the widgets disappear and a note is added to the notebook.

Character widgets: represent a character in the scenario. When the user interacts with them a window appears with the possible dialogs with that character.

Inventory items: the inventory items are the way the inventory objects game elements are represented while in the inventory. They are represented by a name a description and an image.

Inventory: the inventory is the place where the user can see the collected items and their details.

Notebook: the notebook is the place where the user can see the taken notes and their content.

Beginning: At the start of the game a window is displayed showing the title of the story, an introductory text and the objectives of the user.

Ending: when the user ends the story, a window appears with an ending text and the user is redirect to the respective tour after clicking in the ending button. The user can share the experience in a social network or receive a reward pamphlet by giving the email.

Dialogs: there are two views for a dialog: the main character dialog and the NPC dialog.

Main character dialog: this dialog works as if the main character was talking/thinking. It appears only as text in the lower left corner of the screen.

NPC dialog: the NPC dialog shows in a window with the name of the NPC and the image. If there is the need, the interaction dialogs appear.

Events: some actions may trigger some events.

Locked message: this event can happen when a widget is locked to give some feedback to the user. When the event is triggered it shows a dialog in the screen.

Scenario message: this event can happen when the user enters for the first time in a scenario. As same as the locked message event, this event shows a dialog in the screen when triggered.

Appear widget: After performing a certain action a widget (that was hidden) is revealed in a scenario. The action that triggers this event is the interaction with a widget.

3.4.2 Non-Functional Requirements

Performance: when playing games users don't like large loading times. This requires that the front-office is well optimized so that it works smoothly and with good performance. It's desirable that the communication with the back-office is quick and with high speed. Of course there are aspects that don't rely on the platform. One of those aspects is the users' internet connection.

Scalability: this game aims to be a single player game. This means that scalability isn't an issue in this component of the project.

Portability: the front-office can be either played in a web browser or an iPad with internet connection. Nonetheless, it should be kept the graphical stability and easy interaction for either devices.

Usability: Since tourist population is a large one, it involves casual users, meaning that the interface should be of easy comprehension as much as possible. Only the essential operations should be shown to the user and the number of clicks to perform actions should be as less as possible.

3.4.3 Use Cases

After determining what features that belong to the front-office, some of the user options are shown. To represent those options are presented the use cases.

In the figure 3.4.1 shows the model for the front-office. As it can be seen the font-office is composed by the components: Inventory, Notebook, Dialogs, Widgets and Options.

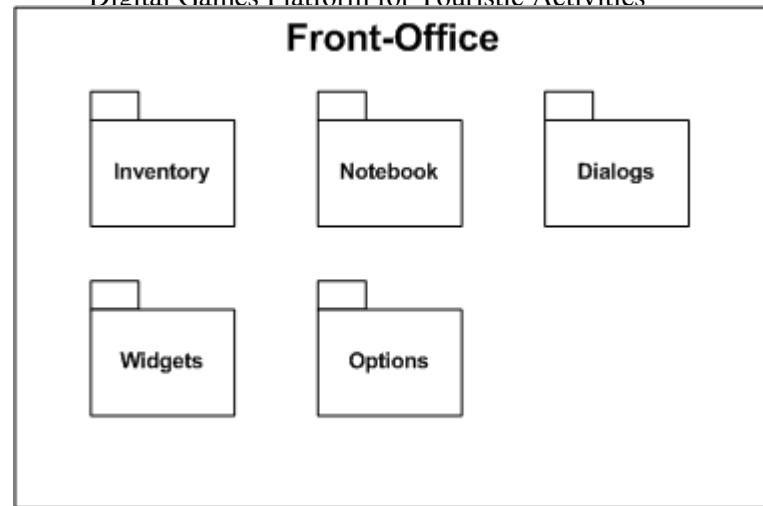


Figure 3.4.1- Front-office use case model

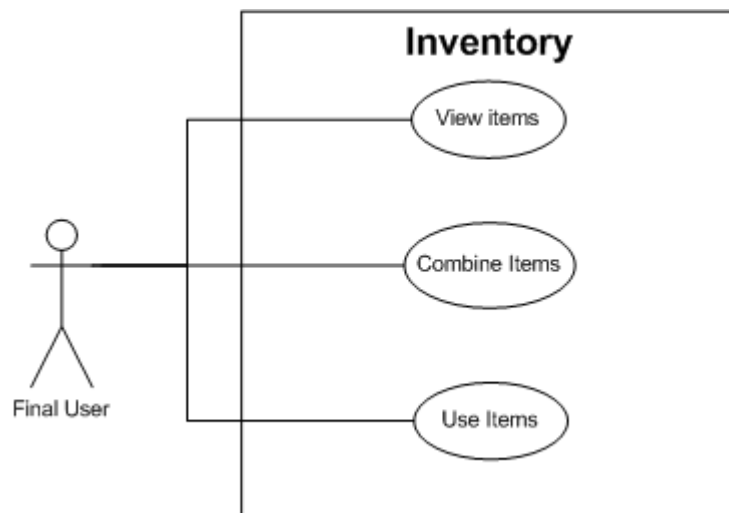


Figure 3.4.2 - Inventory use case diagram

The figure 3.4.2 illustrates the inventory use case diagram. When interacting with the inventory the user can view the respective items. When the user selects a specific item he can combine that item with other items in the inventory or use it in a widget from the scenario.

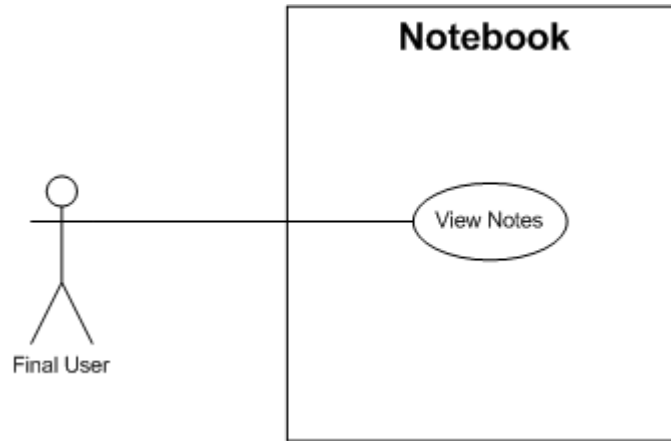


Figure 3.4.3 - Notebook use case diagram

The interaction possible with the notebook is fairly simple, the user can view and examine the content of the notes. The figure 3.4.3 shows that through a use case diagram.

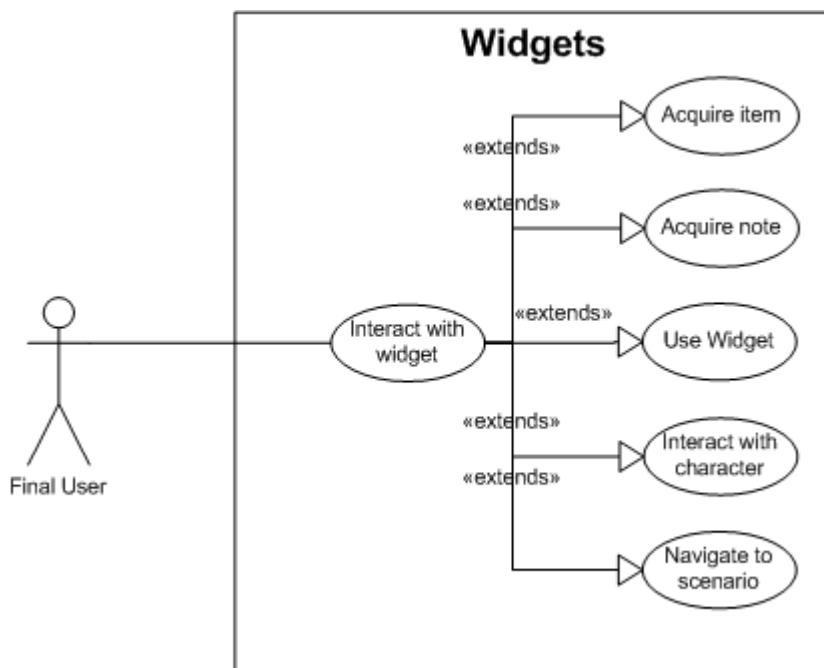


Figure 3.4.4 - Widgets use case diagram

The figure 3.4.4 illustrates the use case diagram for the interaction with the widgets. As we know a widget can perform different actions, depending on its function. The diagram shows us that when a user interacts with a widget, the user can: acquire an item and move it to the inventory, in case the widget is an “inventory object” widget; acquire a note and place it in the notebook, if the widget is a “note” widget; use the widget in case the widget is a “scenario object” widget; interact with a character, meaning the dialogs panel opens, in case the widget is a “character” widget; or navigate to another scenario if the widget is a “navigation spot” widget.

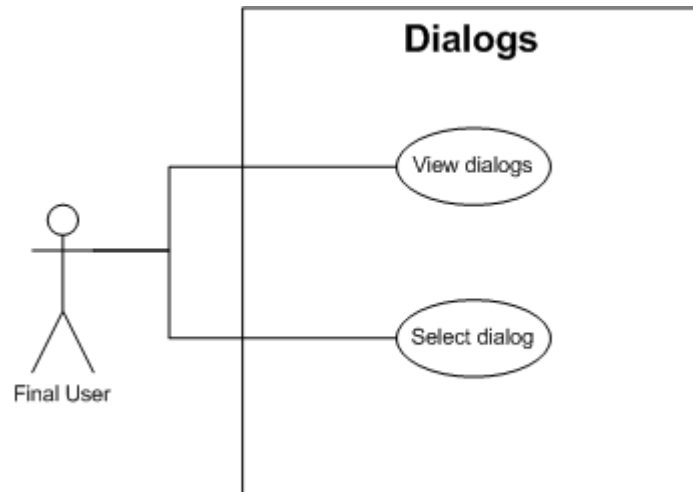


Figure 3.4.5 – Dialogs use case diagram

When interacting with the characters a dialog panel is shown. In that panel the user can view the dialogs from the character that he's talking to. The user can interact with the character by selecting premade dialogs, that the character will give feedback. The dialogs interaction can be seen in figure 3.4.5, the dialogs use case diagram.

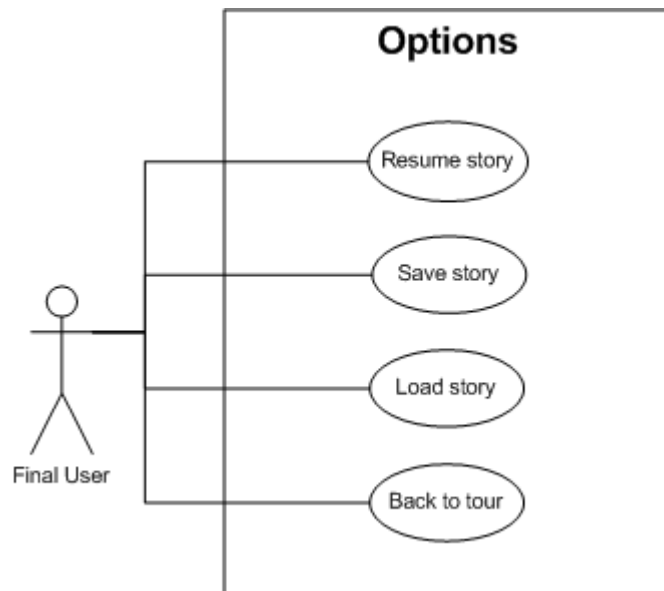


Figure 3.4.6 – Options use case diagram

When interacting with the options panel the user has multiple choices. One of them is to go back to the story that he is currently playing, the “resume story” option. If the user is tired of that story and wants to quit he can go back to the tour that was seeing before he started the story. If the user wants to save the current story so that he can access the data later he can do so in the “save story” option. To access the previously saved data the user can use the option “load story” and continue the story from the point where he left when saved the game. The figure 3.4.6 translates that interaction, with the options use case diagram.

3.4.4 User Interfaces

The user interface should be simple and straightforward since it’s aimed for a casual user population. The interface shouldn’t require a large number of clicks for the user to obtain feedback, if the user has many options he might get confused. The interface should also follow the aLivePanoramics appearance, so that the user doesn’t feel dislocated when entering the Story mode.

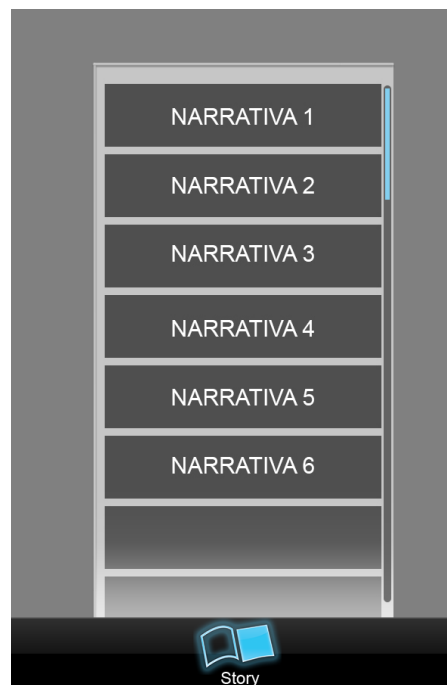


Figure 3.4.7 - Bottom bar story button and stories available menu

To enter a story, the user must choose from the stories available of the tour. To do this, when the user is in one of the other modes, there’s the addition of a new button to the bottom bar. This button allows seeing the list of stories that are attached to a tour and has two different appearances for when it’s active and inactive. If the tour doesn’t have any story available the button doesn’t appear. When the user interacts with the button a menu with the stories available becomes visible. When choosing one of the stories available the user is automatically driven to

the corresponding story. The interface can be seen in figure 3.4.7 where the story button is active.

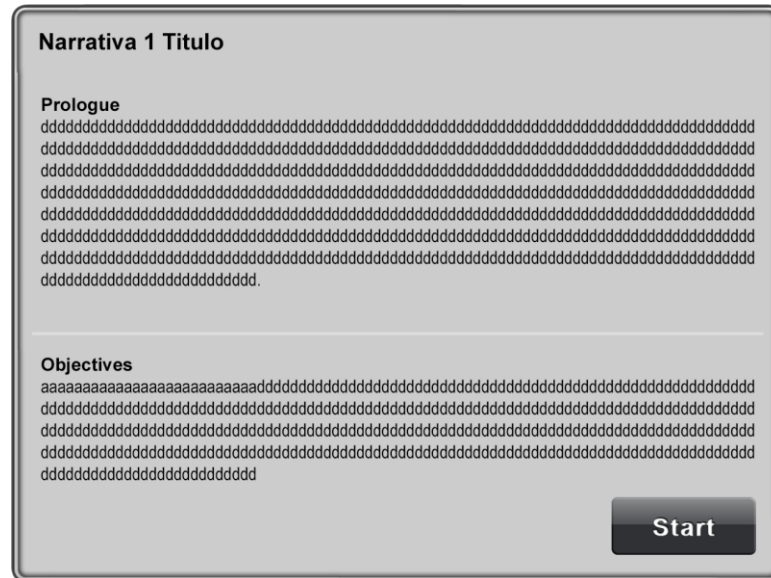


Figure 3.4.8 – Initial panel

When the user enters a new story the initial panel (Figure 3.4.8) is shown. The panel shows the story title, the introductory text and the goals of the user. It has also a start button that allows the user to start playing the game.



Figure 3.4.9 – Inactive notes and inventory buttons

The Story mode adds two additional buttons to the bottom bar, which appear in the right side of it. These buttons are the notes and inventory buttons. They have 3 different appearance states: active, inactive and when the user acquired a new note/item. The user can use them to access the notebook and the inventory respectively. The figure 3.4.9 shows those two buttons in the inactive state.

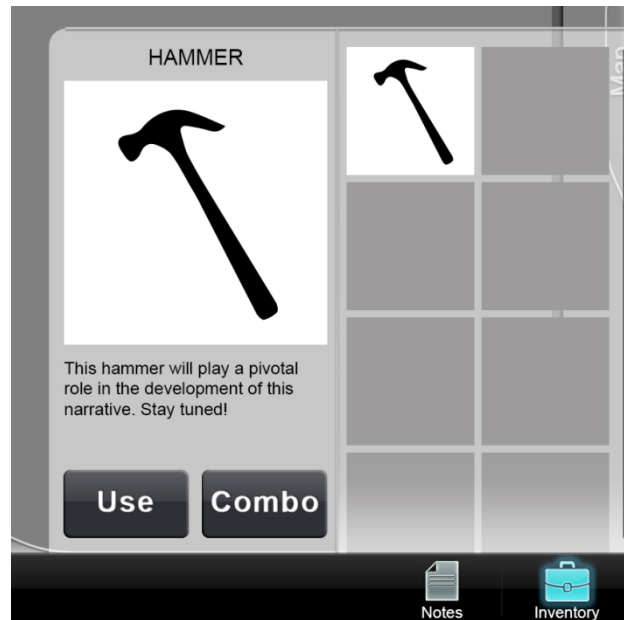


Figure 3.4.10 – Inventory panel and details

The inventory panel is the panel where the user can see the collected items. They appear in a grid and each field has the image of the item that is presented. When clicking on an item in the inventory panel, a panel with the details of the selected item is shown. This panel illustrates the item name and description. The panel also has two buttons, the use button and the combination button. The inventory panel can be seen in figure 3.4.10.

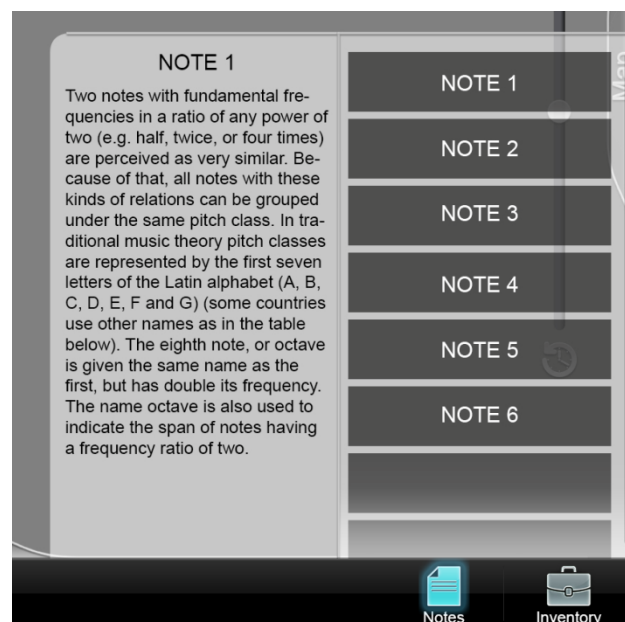


Figure 3.4.11 – Notebook panel and details

The notebook panel is fairly similar to the inventory panel. A list of the notes is shown, showing the title of the notes. When the user interacts on one of those titles, the content of that note is displayed in the details panel. The figure 3.4.11 show how that interface works.



Figure 3.4.12 – Use and combination interface

When the user is trying to use an item or trying to combine it with another item, it is shown in the top right corner of the screen an interface to indicate what item the user is trying to use/combine. As it can be seen in the figure 3.4.12 the interface shows the user what item he is trying to use/combine and what widget/item he's trying to use/combine with.



Figure 3.4.13 – Scenario widget panel

The figure 3.4.12 shows the scenario widget panel. This panel appears when the user is trying to interact with a scenario object widget or a navigation spot widget. The panel shows the name and the description of the widget. To use the widget there's a use button to execute that action.

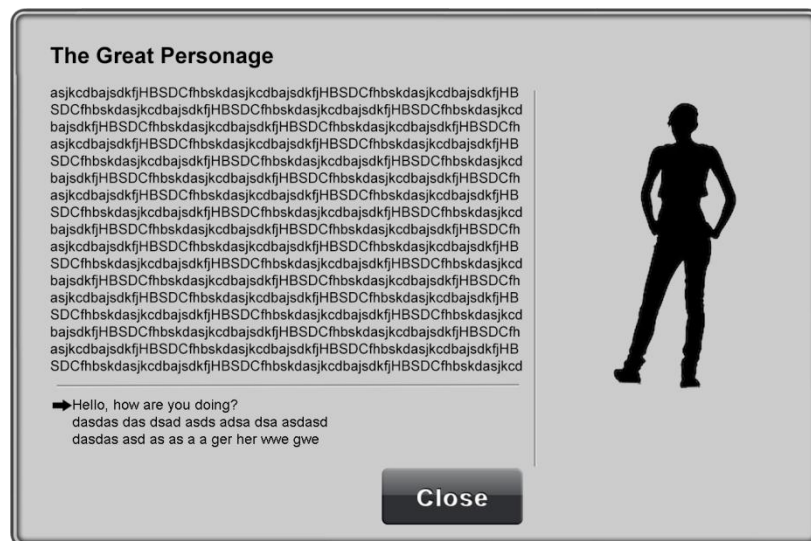


Figure 3.4.14 – Character dialog panel

Besides the main character dialogs that appear as text in the lower right corner of the screen, when interacting with other characters a character dialog panel (figure 3.4.13) appears showing the name of the character and the image. If there is more than one dialog from the character, the dialogs available to choose are shown in the lower division of the panel. When the user has talked everything he wanted with the character he can close the panel using the close button.

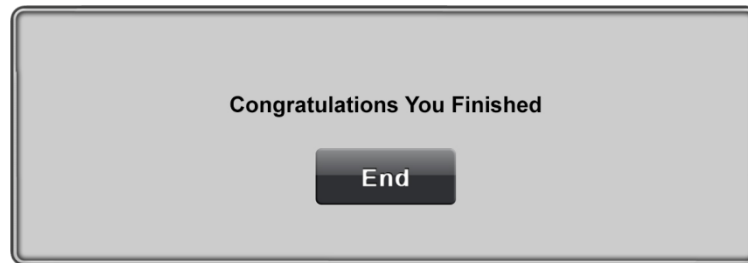


Figure 3.4.15 – Ending panel

The ending panel is quite simple. It shows an ending text and has a button to end the story and go back to the tour. The panel is shown in the figure 3.4.15. The figure is not showing the share and reward buttons.

3.5 Back-office

The back-office is the module where the client can create and manage the stories. This subdivision shows the design of the back-office and how it works.

3.5.1 Functional Requirements

The functional requirements of the back-office are listed below. These are the requirements that are necessary for the back-office to work properly.

Create, update and delete stories: when exploring tours in the back-office, the client can add stories by selecting the name and its description. The client can add as many stories as he pleases. After creating the story, the client can update it, adding other story essentials, such as objects and can as well update the story name and description. If the client doesn't want a certain story anymore, he can always remove it by deleting it.

Create, update and delete characters: the client should be able to create characters. This characters will be able to be used in the in the stories which have a name and an image to represent them. The client can update or delete the created characters.

Add, update and remove tour scenarios: the client can add panoramas that belong to the tour that the story was created in, which are called scenarios in the Story mode. When adding, the scenarios will have the same name as the panorama, but the client can update it, as well as the scenarios' description. If the client doesn't want a certain scenario to belong to that story he can remove it as well.

Create, update and remove inventory items: before adding inventory objects to the scenarios the client has to define the inventory items that belong to the story. The client can also create items that are the result of combinations. The client can choose to create an item from scratch or choose to select one from a list of pre-existing objects. An item is composed by the name, the description, an image and, if it unlocks any game element. The items can be updated and deleted.

Create, update and delete combinations: the client can add combinations to the story. To do so, he must select the two items that combine and choose a third item as a result of the combination.

Add, update and delete inventory objects: if the client wants to place an item in one scenario he can do so, by selecting a position, if it's locked and what inventory item represents it. The client can change the details of the object if he wants to, as well to remove it from the scenario.

Add, update and delete scenario objects: the user can add scenario objects to the scenarios, by choosing its position, name, description, image, if it's locked and if it unlocks a game element. The update and deletion are also available for the scenario objects.

Add, update and delete navigation spots: the client can add to the scenarios navigation spots. The client has to specify the position, the name, the description, if it's locked and for what scenario it travels to. The client can also update those fields anytime or delete the navigation spot.

Add, update and delete notes: the client can add notes to the scenarios. The notes are composed by a name, a description, the content and if it's locked. The client can update those fields anytime and delete notes.

Add, update and delete scenario characters: the client can place characters in the scenarios. To do so, he must choose what character represents from the list of characters, and create the dialogs that belong to the character.

Events: to make the story more interesting the client can add events to it.

Create, update and delete locked dialogs: the user can create "locked dialogs" to give the user more feedback on why a certain game element is locked. To do so, the client must select the character that is going to perform the dialog (the main character is an option too), and what is the message the dialog is going to execute. The client needs as well to choose the game element that is locked and displays the dialog. If the client is unsatisfied, he can update or delete the locked dialog.

Create, update and delete scenario dialog: this event is fairly similar to the "locked dialog", the difference is that the client has to choose the scenario where the dialog is going to be displayed instead of the game element.

Create, update and delete appear game element: if the client doesn't want a game element to appear in the beginning of the story he can do so by creating an event where he has to choose the game element that is not visible and what game element, when activated, makes it visible.

Introduction: the client has to define an introduction for the story, which is composed by a title, an introductory text and the goals that the user will have to accomplish.

Ending: the ending of a story is composed by an ending text. The client has also to decide the ending of the list of possibilities. The possibilities are when entering a new scenario or when obtaining a certain item. Depending on the ending condition the client has to choose the scenario where the story ends or the item.

Create XML files: based on the settings of the stories, the back-office should create XML files for each user.

3.5.2 Non-functional Requirements

Performance: the back-office can be accessed by several clients at the same time. As so, it is important that the back-office is effective and quick enough to answer the clients input. Of

course this factor is affected by external aspects such as the bandwidth of the client and the server machine, but the structure of the back-office should be as much as possible optimized to reduce the waiting time.

Scalability: knowing that the back-office can be accessed by multiple clients at the same time, the system should be scalable enough to cover this.

Security: the stored data of each client should be kept to them and not be accessed by other clients or external users.

Usability: the interface should be familiar to the client, meaning that it shouldn't differ much from the rest of the back-office. It should also be of easy comprehension, and simple to create new stories.

Portability: The access to the back-office can be made via any web-browser, meaning it is required an internet connection to access it. The back-office also has access to the database where the information is stored.

3.5.3 Use Cases

This subsection has the purpose of demonstrating the interaction of the client with the back-office through use cases.

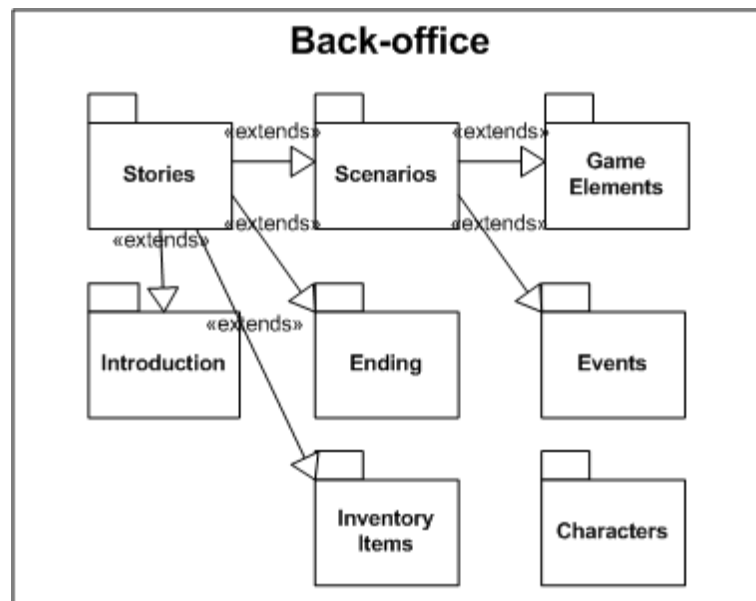


Figure 3.5.1 –Story mode back-office use case model

The back-office is composed by the models of Stories, Introduction, Ending, Scenarios, Game Elements, Events, Inventory Items and Characters. The back-office use case model and its sub-models can be seen in the figure 3.5.1. The dependency from model to model is shown as well. The stories can be composed by an introduction, an ending, inventory items and scenarios. The scenarios on the other hand are composed by game elements and events.

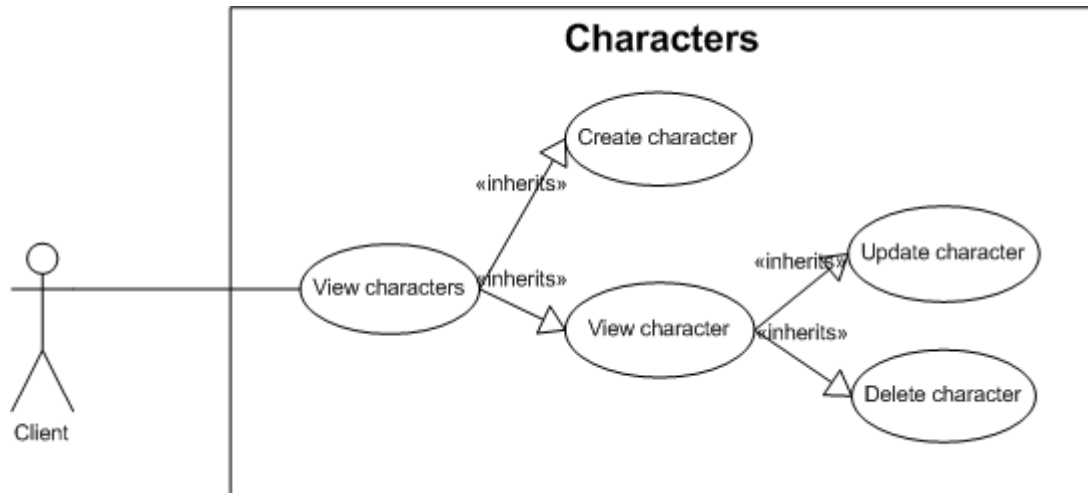


Figure 3.5.2 – Characters use case diagram

Before viewing any specific story the client can view the characters that can be used when creating stories. While there, the client can create a new character or view the detail of the existing ones, update them or delete. There are some default characters that the client can use in the stories but can't update them or delete.

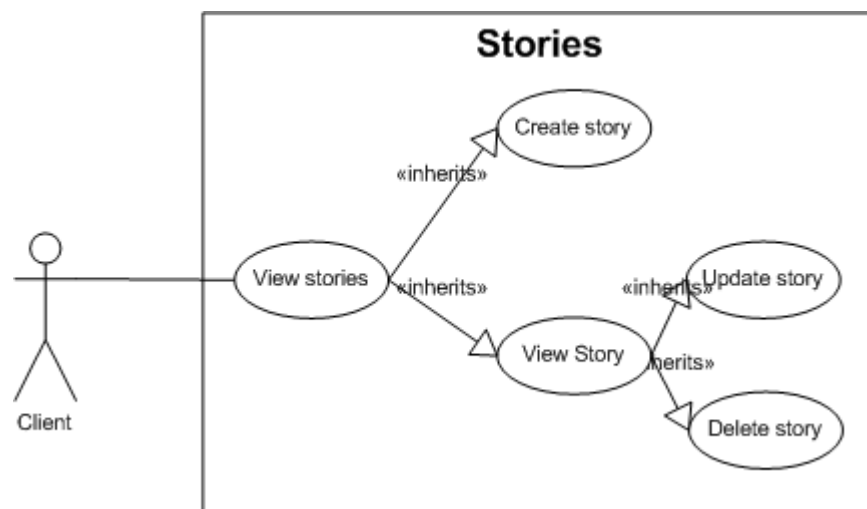


Figure 3.5.3 – Stories use case diagram

The client can view the stories that belong to him. From there, the client can create new stories if he wishes to do so, or view one of the existing stories. When viewing a story the client

can access to the stories sub-models and can as well update or delete that story. The figure 3.5.3 shows the interaction between the client and the stories module.

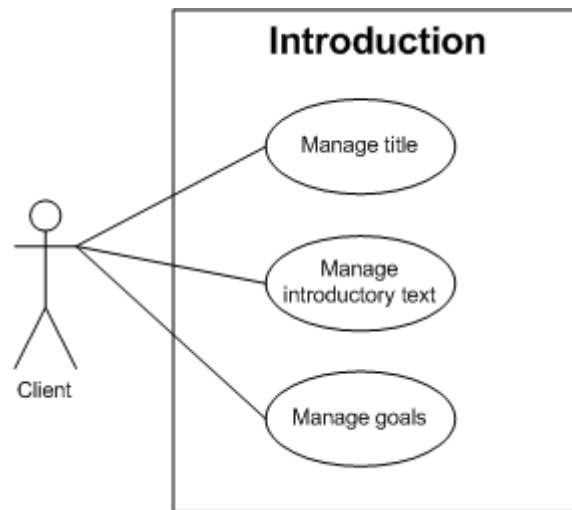


Figure 3.5.4 –Introduction use case diagram

The introduction module is rather simple. The client can manage the title, the introductory text and the goals of the initial panel that the final user sees. This can be seen in the use case diagram of the figure 3.5.4.

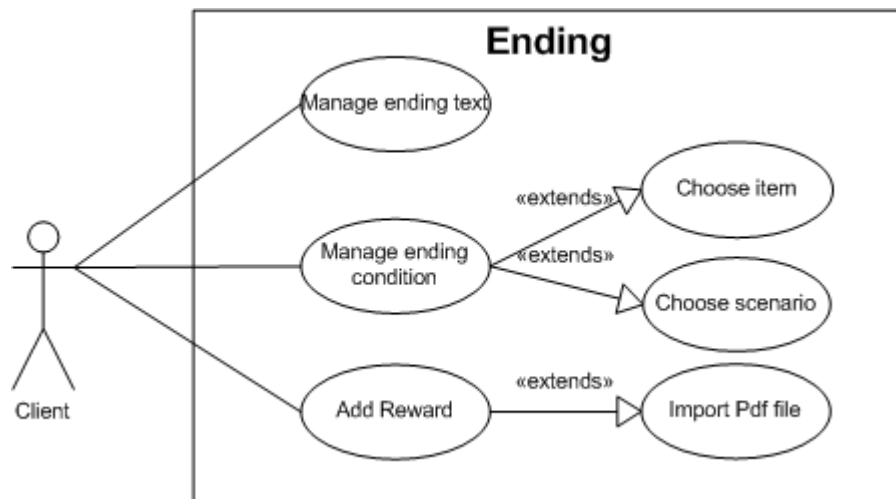


Figure 3.5.5 – Ending use case diagram

In the ending module, besides managing the ending text, the user can also choose the ending condition for the story. Depending on the condition he chooses, the client has to define the item, or the scenario that will end the game. In addition the client can import a Pdf file that will work as a reward. If nothing is imported, in the end of the story there won't be any type of reward. The figure 3.5.5 shows the ending module.

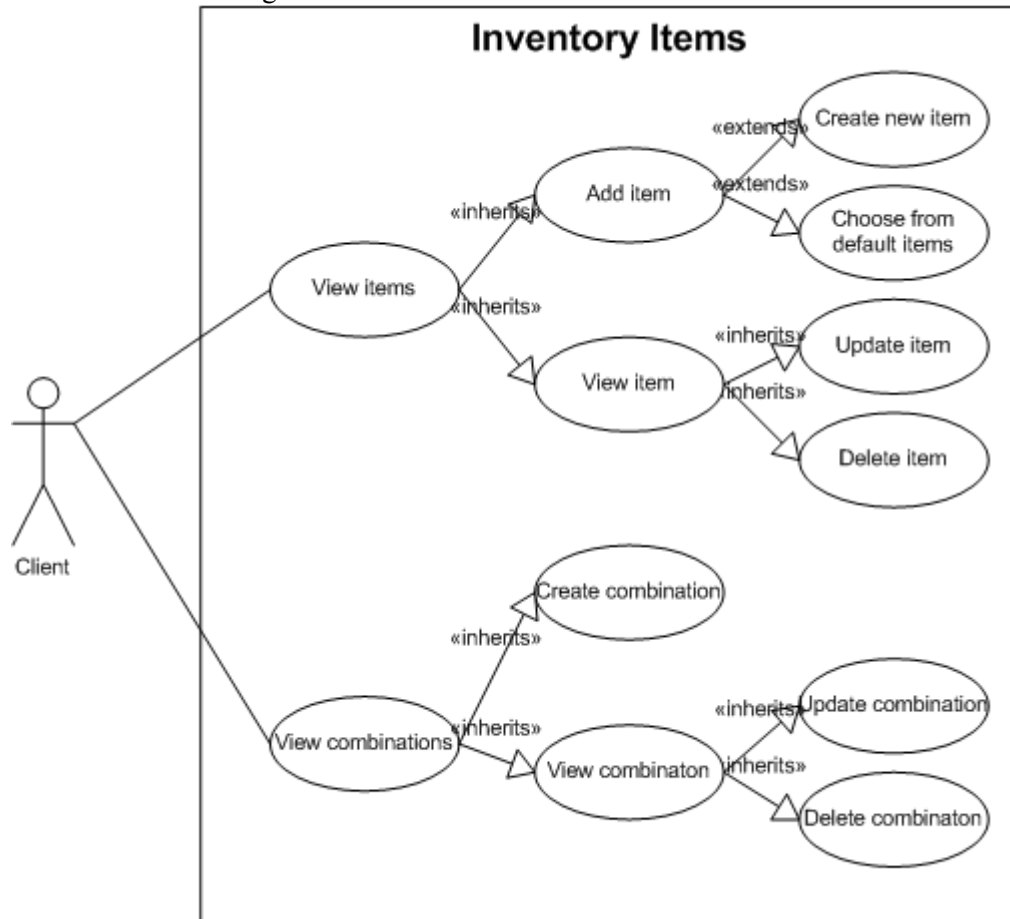


Figure 3.5.7 – Inventory items use case diagram

The figure 3.5.6 shows the inventory items use case diagram. This module includes the management of items and combinations for a certain story. The client can view the items that belong to a certain story and add new items. To add new items the client can either choose to create a new item from scratch or choose to add an item to the story from a list of default ones. The client can as well view the details of the items that belong to the current story, and can update or delete them. While viewing the combinations the client can create combinations of items. The details of these combinations can be viewed, updated and deleted as well.

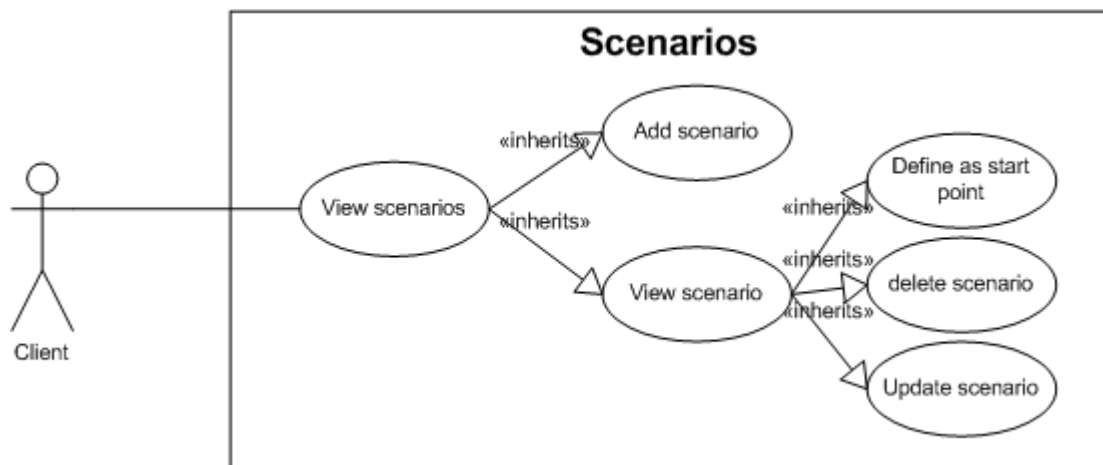


Figure 3.5.6 – Scenarios use case diagram

The scenarios module is where the client can add new game elements and events. Besides that, it's supported the addition of scenarios that belong to the tour. The scenarios can be removed from the story and updated. The client can also define which scenario is the starting one as it is shown in the figure 3.5.7.

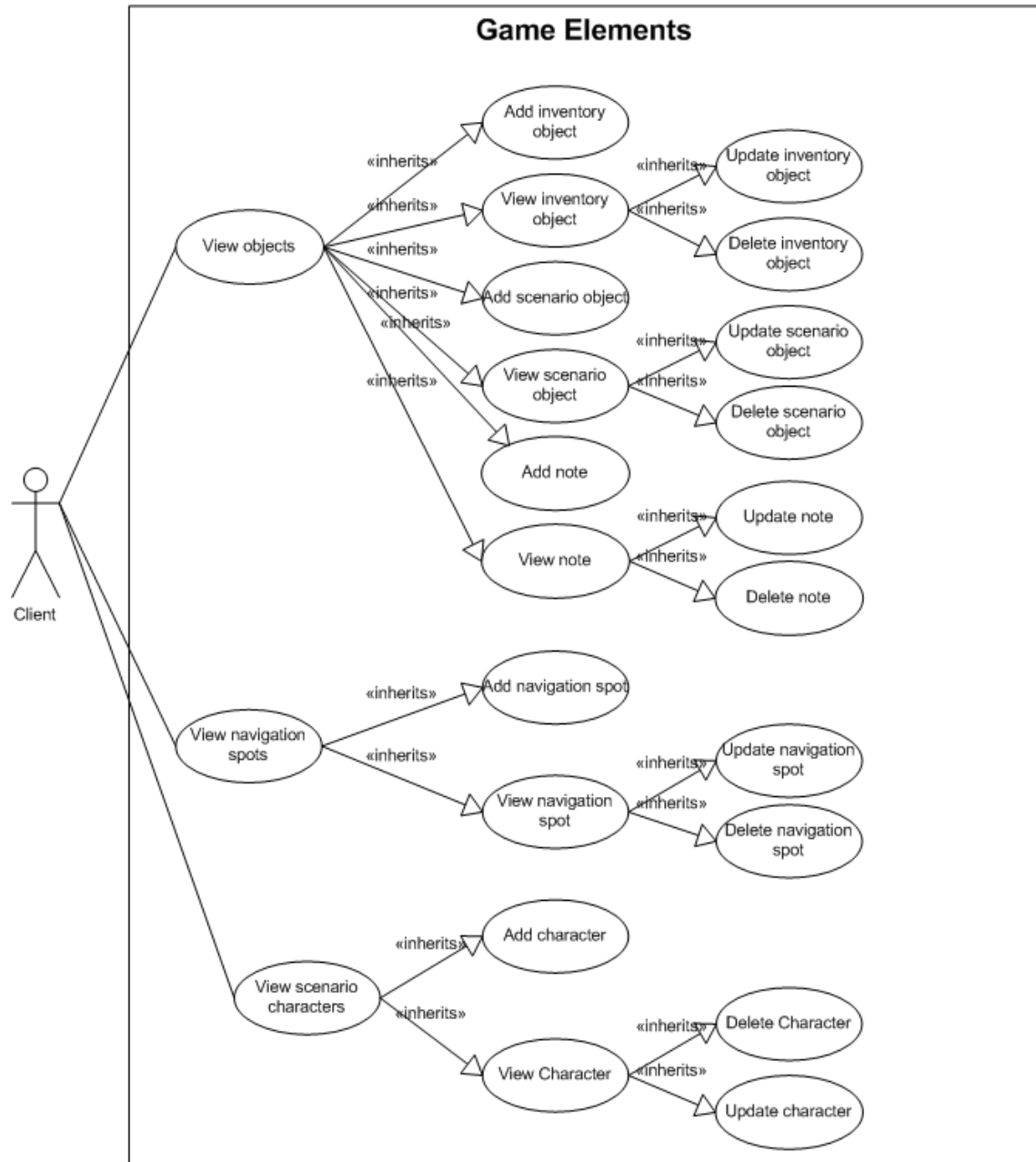


Figure 3.5.8 – Game elements use case module

The game elements module can be divided in three major components: the objects, the navigation spots and the scenario characters. The client can add new objects to the scenario trough these module. The types of objects that the client can create are: inventory objects, scenario objects and notes. The client can also consult the details of these objects and update them. If the object doesn't belong to that scenario or the client isn't satisfied with it, he can

always delete it. The navigations spots are what make it possible to travel from scenario to scenario. The client can view the available navigation spots as well as adding as much navigation spots to the scenario as he pleases. Updating and deleting navigation spots are also available. It is also in the game elements module that the client can add, update and delete characters to a scenario. The interactions of the client with this module can be seen in the figure3.5.8.

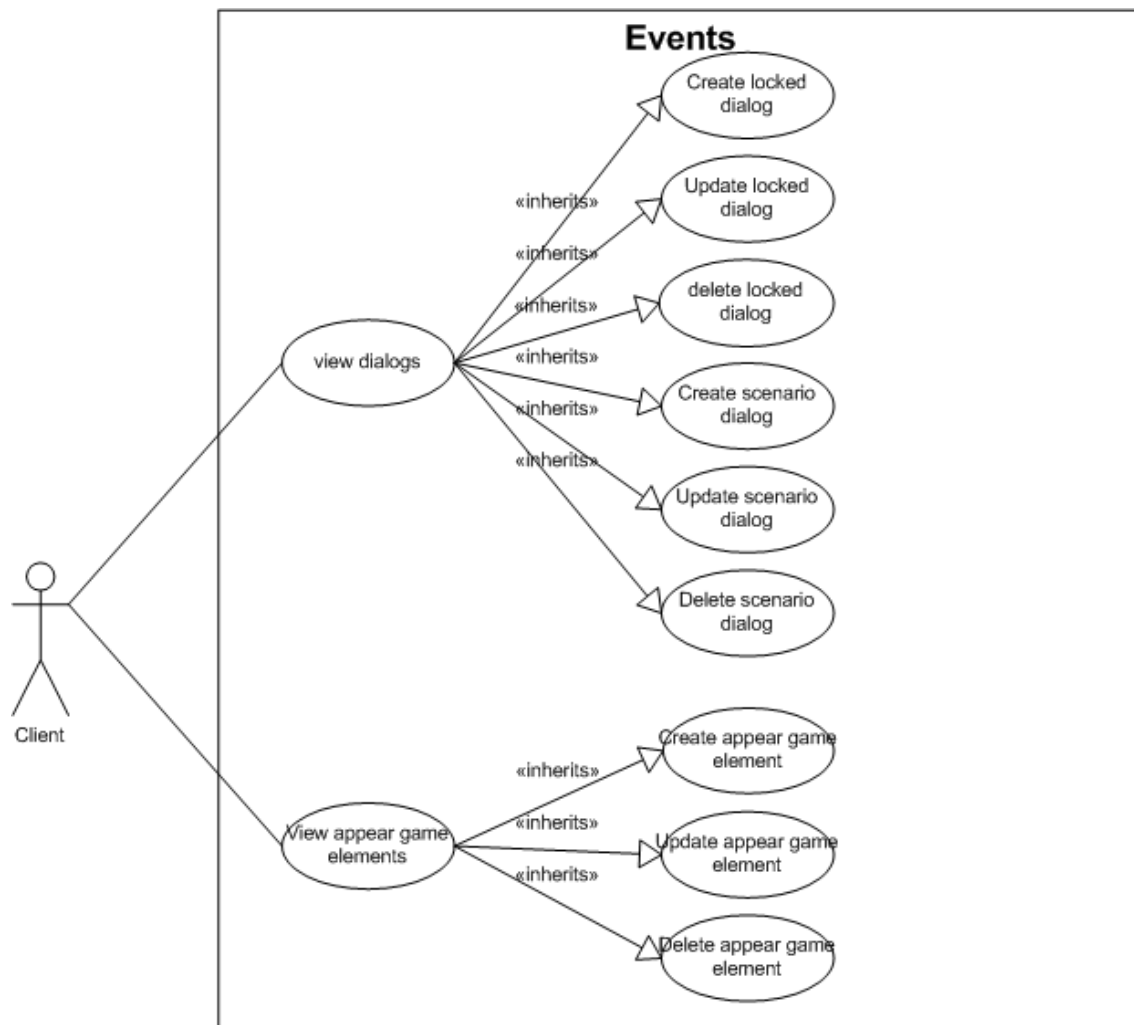


Figure 3.5.9 – Events use case diagram

The figure 3.5.9 shows the interaction with the events module. As it can be seen the client can manage “dialog” events, or “appear game elements” events. In the dialogs, the client can create, update and delete “locked dialogs”, as well to create, update and delete “scenario dialogs”. For the “appear game elements” component, the user can create, update and delete them as well.

3.5.4 User Interface

As in the front-office, the interface in the back-office shouldn't differ much from the aLivePanoramics interface so that the client doesn't feel displaced. In this sub-section are shown some of the most important mockups of the back-office, for the understanding of the concept behind it.

The delete actions in the back-office work as a pop-out window that asks for the clients' confirmation and has two buttons, one to confirm the other to cancel. The update interfaces are similar to the create interfaces, but with restrictions to some fields, if there's the need to.

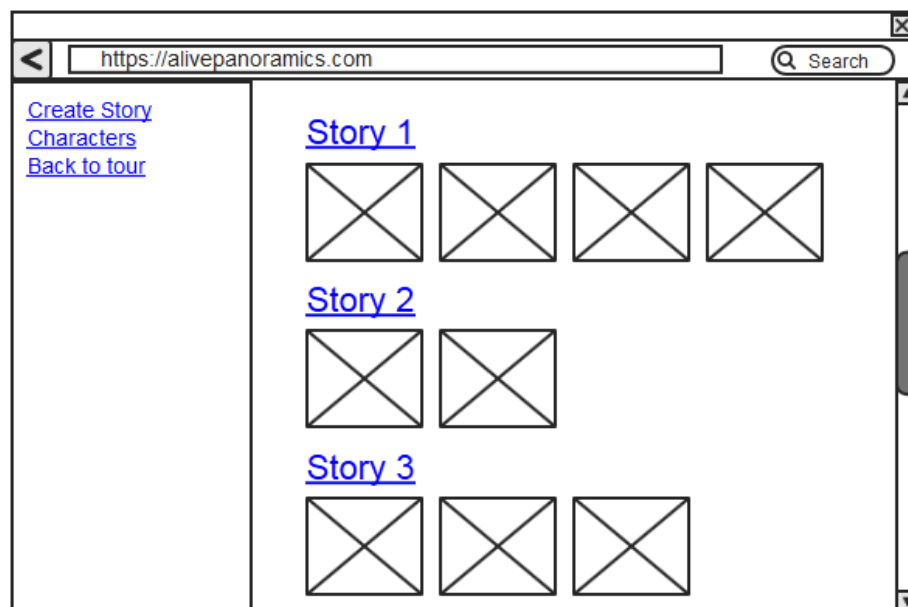


Figure 3.5.10 – Initial page of the story mode

When entering the Story mode in the back-office the client can view his current stories. Each story is followed by the scenarios that represent it. By clicking on a story the client is moved to the story page where he can manage it. Through the menu the client can go to the view of creating new stories, manage the characters that can be used in the stories and go back to the tour, that the stories displayed belong. The interface of this view can be seen in the figure 3.5.10.

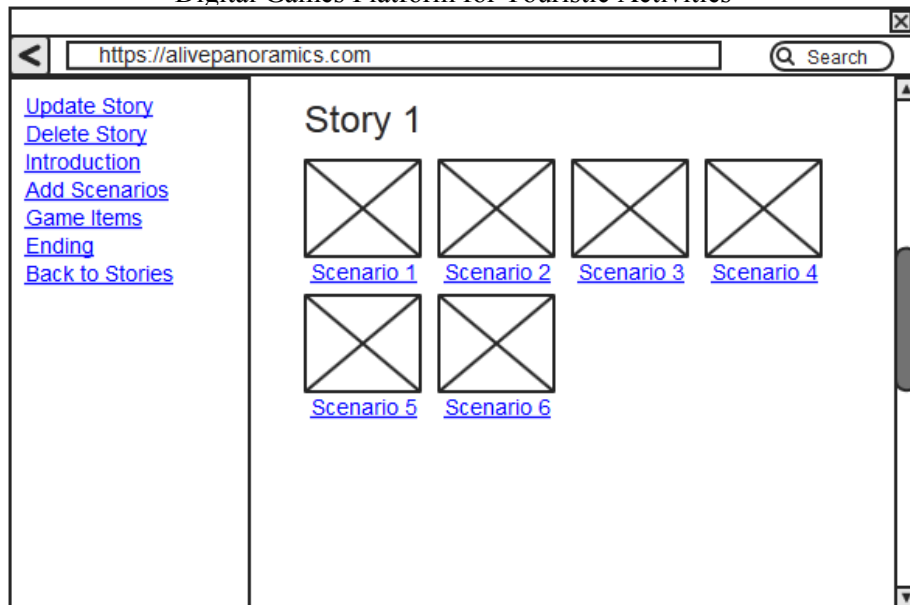


Figure 3.5.11 – Mockup of the story page

The figure 3.5.11 shows the story page of the back-office. As it can be seen, this is the place where the client can manage all the aspects of the story. The client can go to the page to manage a scenario by clicking on one of the images that represents them. Through the menu, besides updating and delete stories, the client can go to the page of the introduction where he can manage the title, the introductory text and the goals text. The menu also allows go to the page to manage game items where can be managed all the items that belong to the game. There is also the ending link, where the client can choose the ending condition and manage the ending text. The menu also allows to go to the page to add scenarios and can go back to view all the stories.

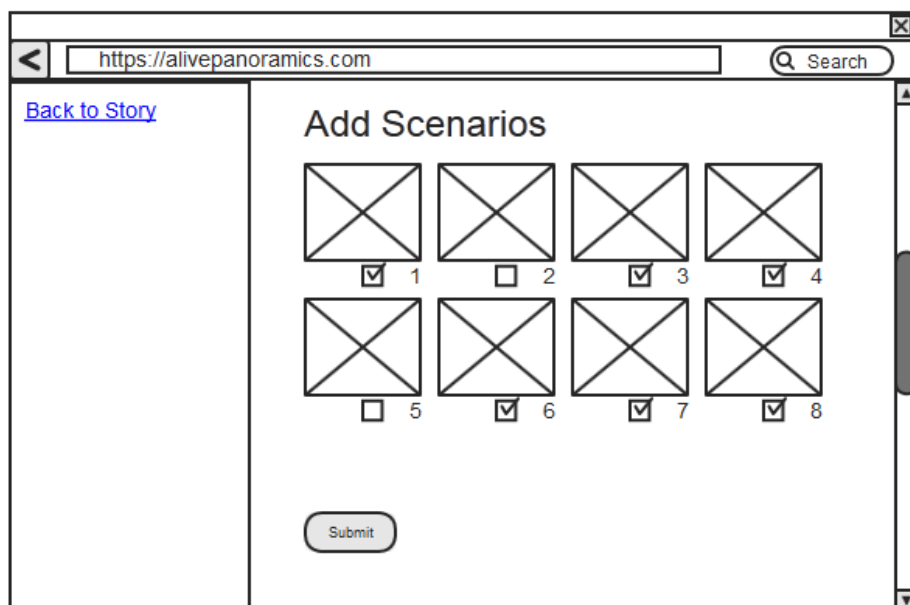


Figure 3.5.12 – Add scenarios page mockup

As we know the scenarios that can be used in a story have to belong to the tours that are associated. For the addition of new scenarios the client can choose them from the list of the panoramas of the tour. As it can be seen in the figure 3.5.12 the client has checkboxes that he can select or unselect if he wants a certain scenario to belong to the story. The menu allows going back to the story without saving the changes.

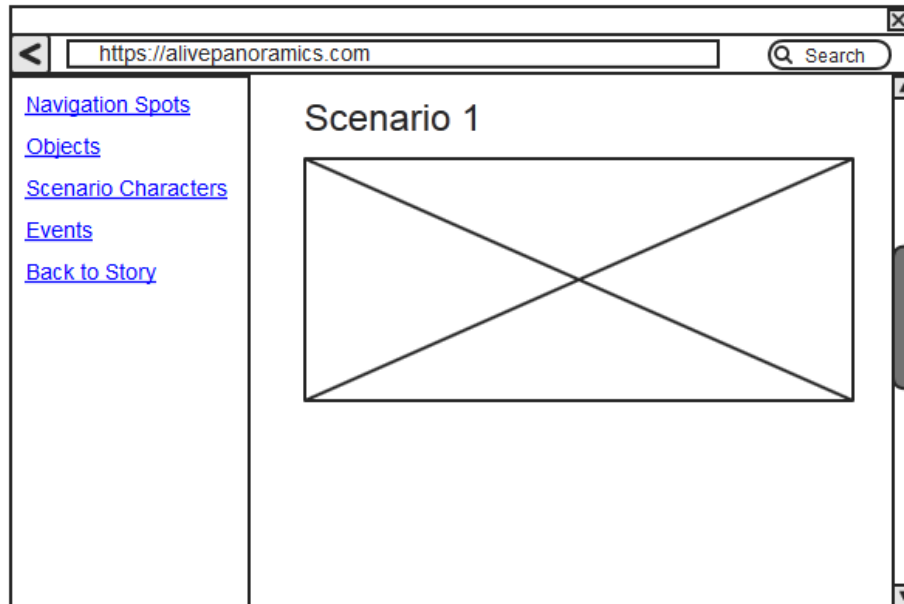


Figure 3.5.13 – Scenario panel mockup

The scenario panel (figure 3.5.13) allows the client to have a better view of the image that represents it. Through this panel the client can manage all the game elements of the scenario and the events as well.

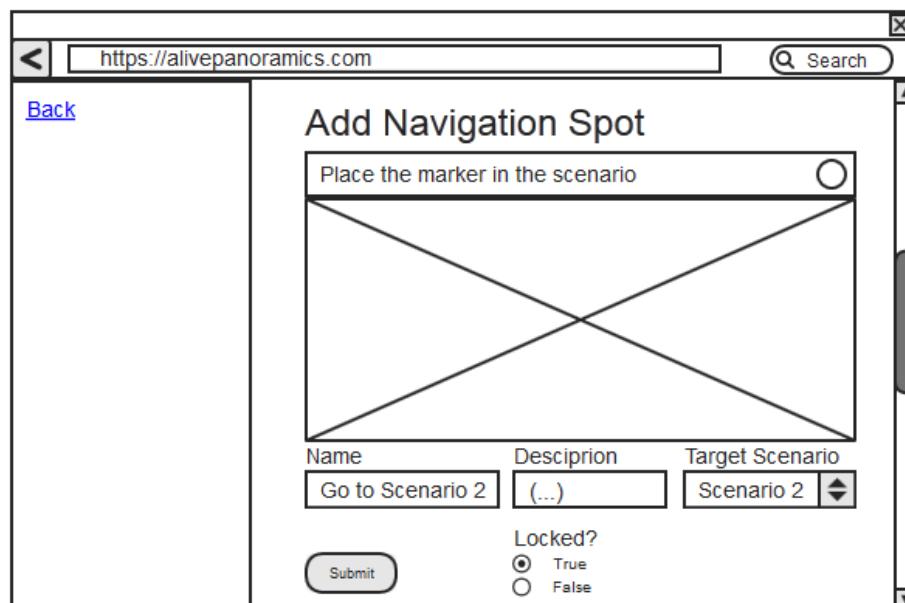


Figure 3.5.14 – Add navigation spot mockup

The figure 3.5.14 shows the “add navigation spot” page mockup, but the pages to add game elements are fairly similar to each other, only the fields change according to the game element. The client has to place a marker in the scenario that will represent the position of the navigation spot. After that the client has to fill some fields. Some of the fields are mandatory or have constraints, for those fields the system shows a warning message and the client can’t submit until the errors are fixed. The page has also a submit button and in the menu there’s a back link in case the client doesn’t wants to save the navigation spot.

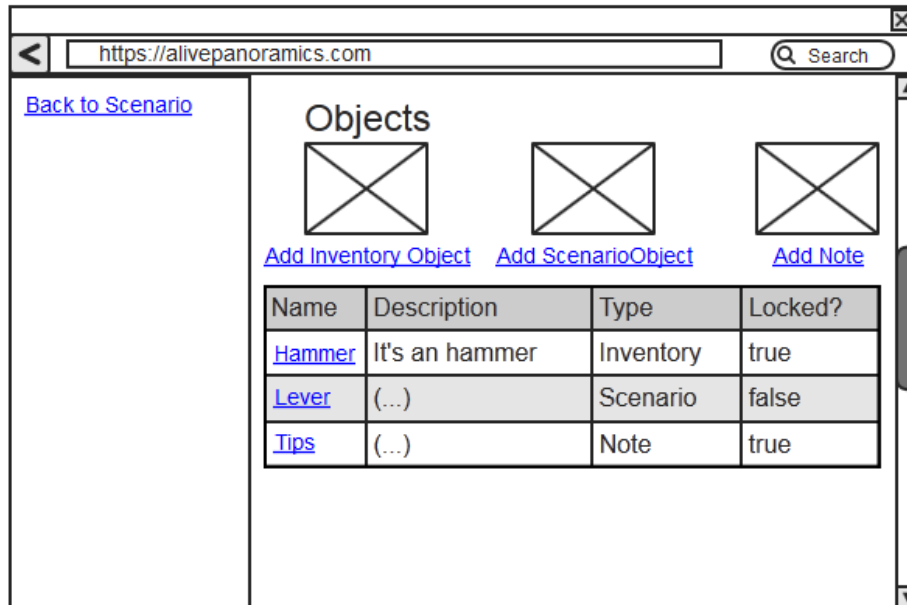


Figure 3.5.15 – Objects page mockup

The objects page (figure 3.5.15) allows the client to manage the objects in a certain scenario. The client can add inventory objects, scenario objects and notes. After adding objects, the client can view them listed in this page. To access their information and details the client can click in the name of the objects.

3.6 Summary and Conclusions

As we have seen, this chapter had the purpose to give the concept of the front-office and back-office of the Story mode.

Knowing that the platform allows the creation of adventure games, where the final user has to solve enigmas by interacting with objects and characters, the front-office is the place where the final user can play it. As so, the front-office must be composed by the scenarios, game elements and events that are defined in the back-office and compose the game. The back-office is the place where the client can create the stories adding the objects that he wants. The interface in both front-office and back-office must be consistent with the one of the aLivePanoramics framework.

Chapter 4

Implementation

This chapter has the purpose to present some details about the implementation and to explain the thoughts behind some of the decisions made. The implementation was applied according to the design mentioned in the previous chapter. As a proof of concept a small demo was made through the platform and put into work.

4.1 Process

Not all of the design architecture was implemented, however the framework allows the creation of stories. There were some constrains which prevented the extension of the project. One of those constrains is the fact that the Augmented reality mode is still in an early development phase, not enabling the possibility of its expansion.

Overall, besides not having the augmented reality implemented, it wasn't implemented scenario characters, the "appear widget" event and the pdf submission for rewards, is not available. Another constraint was the fact that the dialogs aren't interactive, meaning that when a character panel appears, it only displays text.

4.1.1 Database

For the creation of the Story mode, the database accessed by the back-office, had a few changes and some additions.

virtual_tours	
PK	<u>id</u>
	title description created modified user_id

Figure 4.1.1 – Previous virtual tour table

Implementation

Since the Story mode is in fact a Virtual tour, there wasn't the need to create a new table for the story tours. Instead, some modifications were made to the “virtual_tours” table (figure 4.1.1). The “virtual_tours” table has got a title, a description, the date when was created and modified, and a user identification, representing the client that the tour belongs. Two new columns were added to the table: the “is_story_tour” column, which is of the Boolean type and identifies if the tour is a Virtual tour, in case the value is “false”, or a Story tour, in case the value is “true”; the second added column was the “parent_tour_id” that is an identification of the Virtual tour, that the Story tours belong.

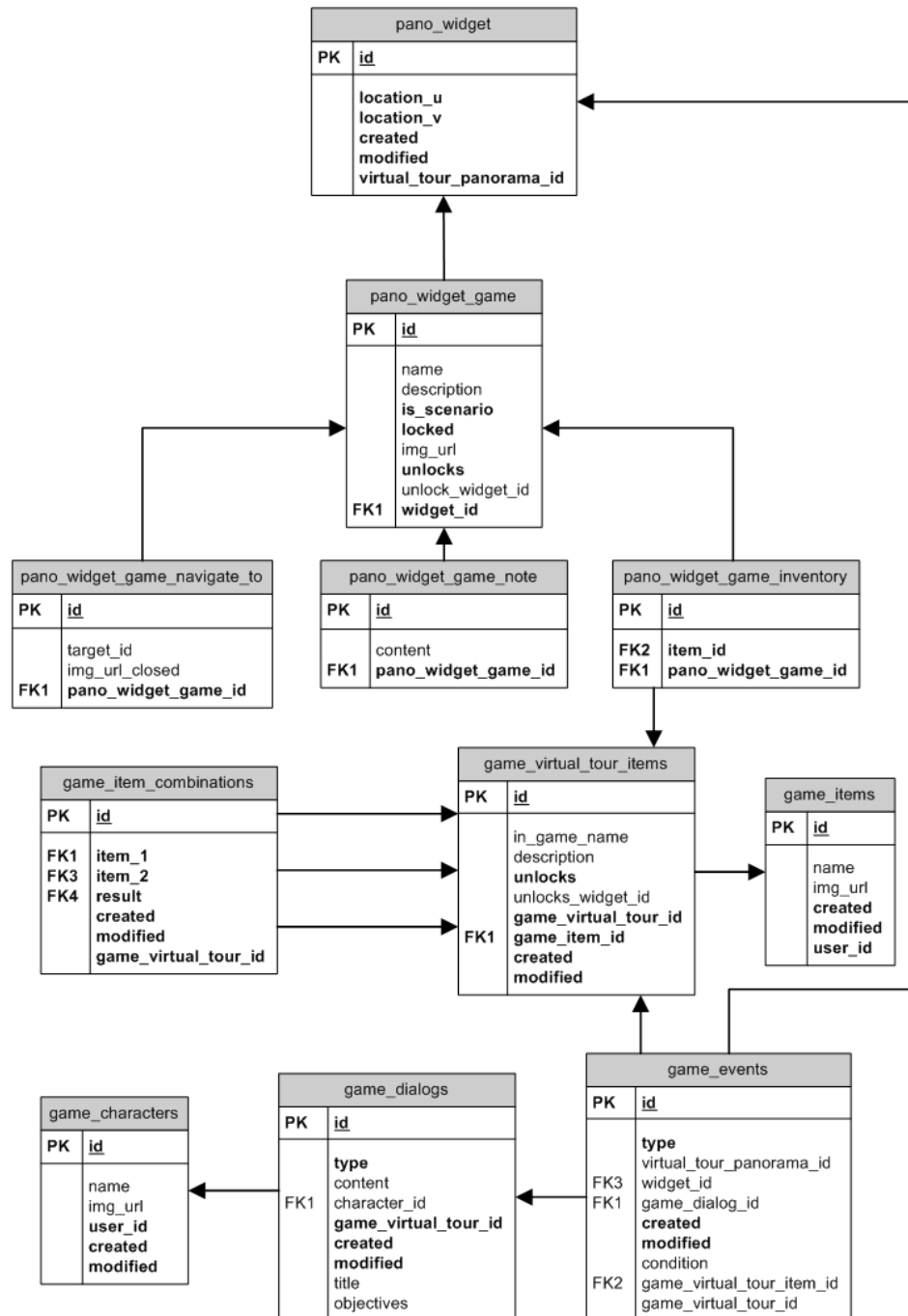


Figure 4.1.2 – Components of the Story mode database

Implementation

The figure 4.1.2 shows the added tables to the aLivePanoramics database. All the important components of the aLivePanoramics database have the “created” and the “modified” columns, that represent the date of when the component are created and modified, respectively. The columns in bold mean that they cannot be null. The “pano_widget” table is a pre-defined table of the aLivePanoramis that represents the widgets, and are connected through the “virtual_tour_panorama_id” to a panorama (scenario) and the respective tour. The “location_u” and “location_v” columns what represents the position of the widget in the panorama.

For the database to store the information of the game widgets, a new child was added to the “pano_widget” table, the “pano_widget_game”, which stores the general information of the game widgets. To avoid many join queries the “scenario object” widget columns were included in this table. As it can be seen, all the widgets have a name, a description, can be locked, have an image and have a “widget_id”, which indicates the father id of that game widget. For the “scenario object” widgets, there is also the inclusion of the “is_scenario” column, of the type Boolean which indicates if the widget is or isn’t a “scenario object”. The “scenario object” widgets also take in consideration the “unlocks” column of the type Boolean as well, which indicates if the widget unlocks another one. In case that variable is true, the id of the widget to be unlocked will appear on the column “unlock_widget_id”.

To store the information about the other three implemented widgets, other three tables were created in the database, which are children of the “pano_widget_game” table. The “pano_widget_game_navigate_to” represents the “navigation spot” widgets and have a “target_id” that specifies the target scenario. It also haves an “img_url_closed”, which will represent the “navigation spot” widget when it’s locked (the “img_url” in this case will represent the widget when it’s open). The “pano_widget_game_note” have the content of the stored notes. As for the “pano_widget_game_inventory” it has the id of the item that widget contains.

The “game_items” table works as a template for the game objects that appear in each story. It has a name and an image that represents the items, and is associated to a specific client. The “game_virtual_tour_items” table stores the information of the items for each story. Since the items belong to a story tour and not to a specific scenario, the “game_virtual_tour_id” is the identification of a virtual tour and not a panorama. The “game_item_combinations” table also belongs to a virtual tour and stores the information about the possible combinations of one story.

The “game_characters” table stores the characters that the client can use in the stories. The characters have a name and an image to represent them. The “game_dialogs” table not only stores the content of a character dialog, but it also stores other type of text information, such as the content of the beginning and ending texts. It also stores the title and goals that appear in the beginning panel. When the stored information is a character dialog, in case the character isn’t the main character, it is necessary to define the “character_id”.

Implementation

The “game_events” table stores the events that happen in the stories. Not only the “locked dialog” and “scenario dialog” events are stored, but the ending condition is, as well, stored in this table.

4.1.2 Communication between front-office and back-office

As was mentioned before the communication between the front-office and the back-office is made through XML files. To be able to make the front-office recognize the new created widgets and game settings some changes to the creation of old XML files were made and the addition of a new XML file was implemented. The old XML contained the information about the tours and the panoramas. The necessity to distinct the virtual tours from the story tours raised, and the old XML besides the id and name of the tour, now contains the information indicating if it's a virtual tour or a story tour. Moreover, it also contains the information about the new widgets. The new implemented XML contains all the information about the game: which widgets are locked; the possible combinations of items; the items and their details; the characters; the dialogs, including the beginning and ending texts; the events that occur in the game. The front-office reads these XML files and then creates the stories accordingly to the received input.

The need to place the “locked” information in the general settings of the game and not associated with the widget itself arose because a widget is connected to a scenario and the “locked” info is global (for example, a scenario widget can unlock a widget from a different scenario).

4.2 Results

Unfortunately it wasn't possible to get a decent real case prototype. It was intended to be done a prototype for the museum of the Paço dos Duques de Bragança, but due to time constraints the required resources (images, icons, etc) couldn't be prepared in time. Nonetheless, a demo to show the capabilities of the game was done with resources taken from the internet. The demo had the purpose to show the clients the potential of the platform and had a very simple storyline. This demo prototype also allowed doing a “thinking out loud” evaluation to a population of 5 people. Some decent feedback was collected from that evaluation and taken in consideration to change some of the user interface (the presented interface in chapter 4 already takes this evaluation in consideration). Some of the changes that were conducted was the removal of some buttons, in particular close buttons of panels and the examine button that allowed to see the details of the game elements (very few people used this button). These changes allowed the user to get more feedback with less clicks.

Implementation

As said before the story of the demo was very simple and was created for capabilities demonstration purposes. A more detailed structure of the demo game can be seen in the “Stories creation manual”.

In the demo the user has to go through a sequence of scenarios to get to a specific one, the pool. The story of the main character is that he just got out of work and wants to relax. The goal of the user is to get to the pool. The demo has four scenarios the work place, the city, the home and the pool.



Figure 4.2.1 – Beginning panel

The figure 4.2.1 shows the beginning panel of the demo. The title chosen for the story was “Welcome to Game Mode” followed by the introductory text that explains that this story is a demo and has the purpose to show the potentials of the platform. It also explains the context of the story and shows the goals.

Scenario 1 - Work Place:

Game elements:

- Inventory objects: Rod – unlocked; Car key – unlocked; Shovel – locked
- Notes: Tips - unlocked; Lorem Ipsum - unlocked
- Navigation Spots: Go to the city – locked

In this scenario the “Car key” unlocks the “Shovel” and the “Shovel” unlocks the “Go to the city”. Both locked game elements show a main character dialog with details on why the element is locked. The “Shovel” dialog says that the shovel is inside a car and the door is locked, while the “Go to the city” states that the path is blocked by rocks.

Implementation

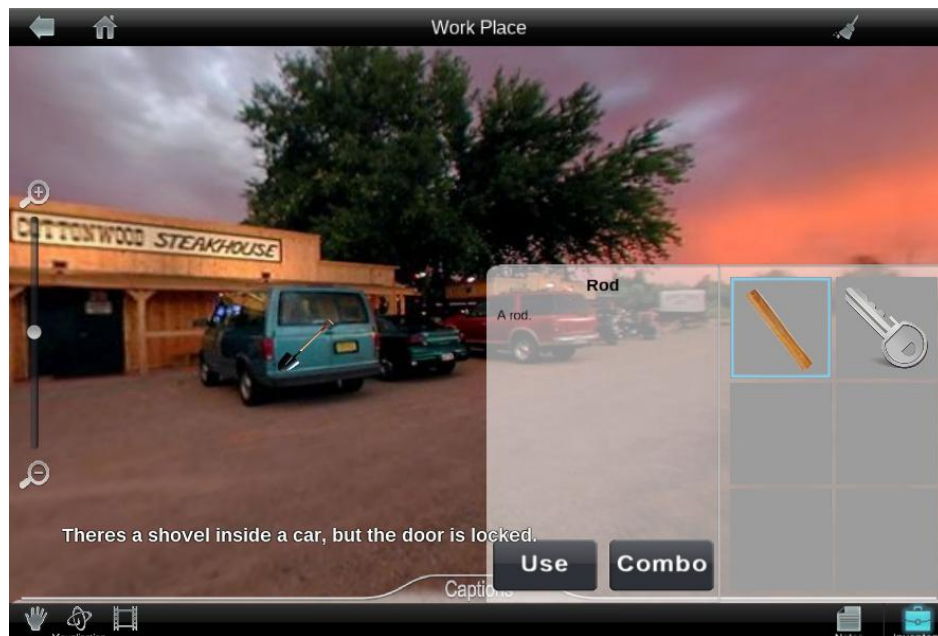


Figure 4.2.2 – “Work Place” scenario and some interface

The figure 4.2.2 shows the inventory details where the user already picked up the “Rod” and the “Car Key”. It also shows a “locked dialog” when trying to pick up the “Shovel”.

After solving the challenges of the first scenario the user can navigate to the second scenario.



Figure 4.2.3 – Character panel that appears after trying to use the “Go to the pool” locked element

Implementation

Scenario 2 – City

Game elements:

- Inventory objects: Hammer head – unlocked
- Scenario objects: Code panel – unlocked
- Navigation spots: Go to the work place – unlocked; Go to the pool – locked; Go home – locked

The code panel unlocks the “Go home” element. When entering the scenario a character panel appears welcoming the user to the city. There are also “locked dialog” events for the locked elements. For the “Go to the pool” element a character panel appears (figure 4.2.3) saying that the user needs to pay to enter, as for the “Go home” element a “main character dialog” appears saying that the user needs to enter a code in the code panel.

Scenario 3 – Home

Game elements:

- Inventory objects: glue – unlocked; Wallet – locked
- Navigation Spots: Go to the city – unlocked

The “wallet” unlocks the “Go to the pool” element from the second scenario. If the user picked up all the possible inventory objects to the moment it will have three objects in the inventory (figure 4.2.4): the “Rod”, the “Hammer head” and the “Glue”. The user can combine the objects and make a hammer with them. The hammer unlocks the “Wallet” element.

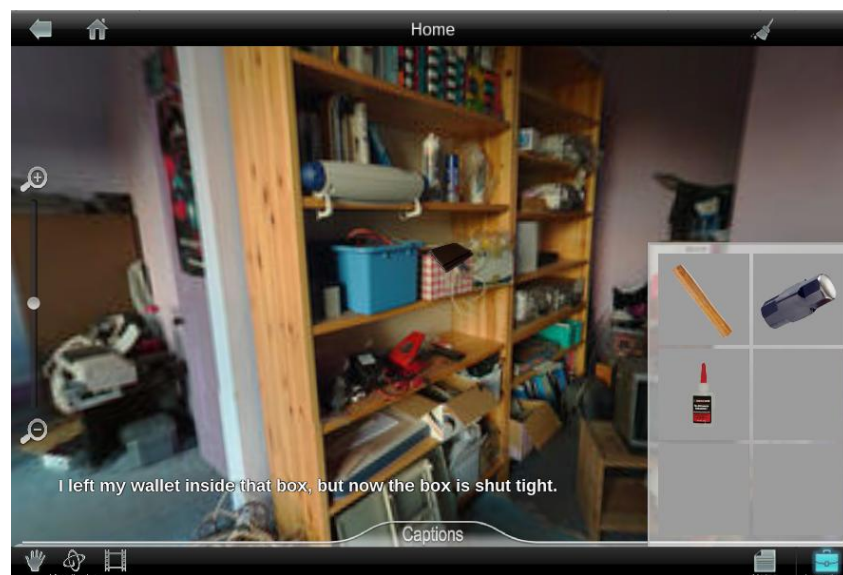


Figure 4.2.4 – The 3 elements in the inventory and a dialog after interacting with the “Wallet”

Implementation

Scenario 4 – Pool

When the user enters the scenario the ending panel appears congratulating the user for finishing the game as shown in figure 4.2.5.

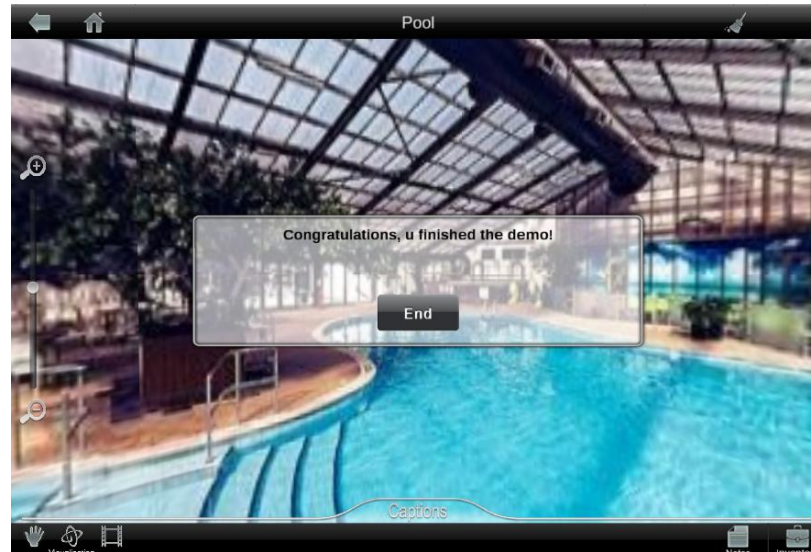


Figure 4.2.5 – Ending scenario

4.3 Summary

Although the platform wasn't fully implemented, most of the important components are present, allowing the possibility to create interesting stories. The changes to the aLivePanoramics framework and database were successfully applied allowing the communication between the front-office and the back-office to be of good quality. A simple prototype was made as proof of concept and to show some of the capabilities of the platform.

Chapter 5

Conclusions and Future Work

This dissertation had three major segments: conceptual, technological and business. The conceptual part mainly focused on contextualizing the project and understanding how to please the target audience. The tourist population is very wide, meaning that involves casual gamers who aren't very familiar with technologies, as so, in the game design definition that was taken in consideration. To simplify the game, an easy to interact user interface was used as well, with familiar terms to the users. Another important context was the fact that the project had to work with the aLivePanoramics architecture, this meaning that it has a front-office where the final users can play the game and a back-office where the clients can create and manage the games. The created games work with panoramas and widgets mainly, because those components are a key factor in the aLivePanoramics framework. The fact that it was mandatory to work under the aLivePanoramics framework, brings us to the technological part of the dissertation. Since the framework was already in development, an extension to the same was required using the technologies used to develop the framework. For the front-office the engine Unity3D with the plug-in NGUI was used, while in the back-office was used the Yii framework. The communication between these two components is made trough XML files and web-services. The dissertation had as well a business part where it had to have some value in the market.

5.1 Goal Satisfaction

The goal of this dissertation was to create a platform that would be able to create digital games for tourism activities. Even though some of the requirements weren't achieved due to time and framework constrains, the platform was successfully created with good results. During the development, a focus in the most important components allowed to achieve a game design where the clients can build a large variety of games with different stories.

Conclusions and Future Work

The platform allows creating adventure games where the final user has to solve an enigma to get to the end. The games are composed by scenarios, which are composed by game elements. These elements can be objects, navigation spots or notes. They can be locked and the final user has to find a way to unlock them. The objects can be fixed in the scenario or can go to the inventory. Both objects can unlock other elements, but the inventory objects can be combined to create another inventory object. The navigation spots allow traveling from scenario to scenario and the notes have some kind of relevant content to the game. Events when an element is locked and when entering a scenario, can be added to the stories, displaying some dialogs from characters. In the ending, a panel is shown to the final users and redirects to the main menu. The possible endings are entering a specific scenario or obtaining an item. In all created games the final user will have with him an inventory and a notebook, where he stores the inventory objects and the notes respectively.

The real case prototype wasn't achieved, although there were some meetings with the clients of Paço dos Duques de Bragança, and the beginning of a story for their museum is in progress. To obtain all the required resources is a slow process and require many people, like photographers, designers and others. Nonetheless, a small prototype to show the capabilities and evaluate the platform was generated. This prototype was also used in the meeting with the clients.

5.2 Future Work

Since this is a platform to create games, there is a lot that can be improved and updated. Besides the requirements that weren't implemented such as the "appear widget" event, the character widget, the submit a pdf file as reward and the sharing with friends option, there could also be implemented sounds (for now only have the default ones for buttons), the possibility of an object to unlock multiple elements, the addition of more events and widgets with other actions.

It is also important to finish the real case prototype, since it proves the liability of the project for real clients and evaluate their satisfaction with the product.

References

- [1] Jacob, João. 2009. A Location-Based Digital Game. Faculdade de Engenharia da Universidade do Porto.
- [2] Bell, Marek, Matthew Chalmers, Louise Barkhuus, Malcolm Hall, Scott Sherwood, Paul Tennent, Barry Brown, Duncan Rowland, Steve Benford, Alastair Hampshire, Mauricio Capra. 2006. Interweaving Mobile Games With Everyday Life.
- [3] Korhonen, Hannu and Elina M. I. Koivisto. 2006. Playability Heuristics for Mobile Games.
- [4] Steiniger, Stefan, Moritz Neun and Alistair Edwardes. 2006. Foundations of Location Based Services. CartouChe - Lecture Notes on LBS. V. 1.0.
- [5] Schiller, Jochen and Agnès Voisard. 2004. Location-Based Services. Elsevier.
- [6] Rahid, Omer, Ian Mullins, Paul Coulton and Reuben Edwards. 2006. Extending Cyberspace: Location Based Games Using Cellular Phones. ACM Computers in Entertainment. Vol. 4. No. 1.
- [7] Nicklas, D., Ch. Pfisterer and B. Mitschang. Towards Location-based Games. Breitwiesenstr. 20-22, 70565 Stuttgart, Germany.
- [8] Jacob, João and António Coelho. 2010. Issues in the Development of Location-Based Games. <http://www.hindawi.com/journals/ijcgt/2011/495437/> . (last accessed on July 20, 2012).
- [9] Sotamaa, Olli. 2002. All The World's A Botfighter Stage: Notes on Location-based Multi-User Gaming. Tampere: Tampere University Press.
- [10] Hall, Justin. 2004. Mogi: Second Generation Location-Based Gaming. <http://thefeaturearchives.com/100501.html> . (last accessed on July 20, 2012).
- [11] Mopius. The Journey II. 2012. <http://journey2.mopius.com/> . (last accessed on July 20, 2012).
- [12] Marques, Fabricio Rosa. 2010. Colonia Mysteria – Exploring pervasive educational games.

[13] GroundSpeak. Geocaching. 2010. <http://www.geocaching.com> . (last accessed on July 20, 2012).

[14] Qualcomm. Vuforia. 2012. <http://www.qualcomm.com/solutions/augmented-reality> . (last accessed on July 20, 2012).

[15] Ballagas, Rafael, Steffen P. Walz and Jan Borchers. April 2006. REXplorer: A Pervasive Spell-Casting Game for Tourists as Social Software.

[16] Lynch, Wesley. 2012. Tourism & the Digital Age. <http://www.exploreonline.co.za/articles/tourism-the-digital-age> (last accessed on January 20, 2013).

[17] 2011. The Best Game Engines For Indie Game Developers. <http://nuverian.net/2011/01/17/the-best-game-engines-for-indie-game-developers/> (last accessed on January 20, 2013).

[18] Jones, Chris. Adventure Game Studio. 2006. <http://www.adventuregamestudio.co.uk/> (last accessed on January 20, 2013).

[19] Visionaire Team. Visionaire Studio. 2012. <http://www.visionaire-studio.net/> (last accessed on January 20, 2013).